Escola Politècnica Superior de Castelldefels
UNIVERSITAT POLITÈCNICA DE CATALUNYA

CHEMNITZ UNIVERSITY OF TECHNOLOGY

MASTER THESIS

TITLE: Initialization algorithms for wireless ad-hoc networks

MASTER DEGREE: Master of Science in Telecommunication Engineering & Management

AUTHOR: Carlos Agreda Ninot

DIRECTOR: Prof. Dr.-Ing. Ulrich Heinkel

DATE: April 26 th 2010

Títol: Algoritmes d'inicialització per a xarxes ad-hoc sense fils

Autor: Carlos Agreda Ninot

Director: Prof. Dr.-Ing. Ulrich Heinkel

Data: 26 d'abril de 2010

Resum

L'objectiu d'aquest *master thesis* és la implementació de models de simulació d'algoritmes d'inicialització per xarxes de sensors que siguin energèticament eficients, així com la seva posterior simulació i evaluació.

Primerament es dona una visió general de les estratègies que es poden trobar actualment per a la inicialització i exploració de xarxas ad-hoc sense fils.

Seguidament es fa una presentació de las característiques i propietats dels algoritmes d'enrutament basats en clusters que han estat seleccionats per a la seva implementació en aquest treball. Aquests algoritmes són LEACH, LEACH-C, l'extensió d'ambdòs per l'aprofitament de la energia solar, HEED i un protocol basat en la transmissió directa implementat només per tenir una referència a l'hora de comparar la resta de protocols.

D'altra banda tots aquests protocols han estat implementats i simulats mitjançant OMNeT++ 4.0, un simulador d'events discrets el qual és distribuït amb llicència de codi obert.

A continuació es presenten les simulacions fetes per tots els protocols amb diferents paràmetres i condicions amb l'objectiu de provar la seva funcionalitat i veure el seu comportament en diferent tipus de xarxes de sensors. Posteriorment les simulaciones dels algoritmes són comparades especialment en termes d'eficiència energètica i en el cost que la communicació comporta en cadascun d'ells. S'hi presenten diferents comparacions entre LEACH i LEACH-C amb les seves respectives extensions solars, una comparació d'un HEED amb paràmetres optimitzats amb un HEED amb paràmetres no optimitzats i una comparació final entre els protocols One-hop, LEACH, LEACH-C i HEED.

Per finalitzar es presenten conclusions sobre el funcionament dels protocols implementats en aquest treball i es donen pautes per a una futura continuació i millora d'aquests. A més, es presenta un breu estudi de l'impacte mediambiental que aquest treball pugui tenir.

Title: Initialization algorithms for wireless ad-hoc networks

Author: Carlos Agreda Ninot

Director: Prof. Dr.-Ing. Ulrich Heinkel

Date: April, 26th 2010

Overview

The aim of this master thesis is the implementation of simulation models and the simulation of energy-efficient network initialization algorithms.

First of all, it is presented a survey of state-of-the-art strategies for network initialization and exploration in wireless ad-hoc networks. Among the routing approaches presented in the survey it has been chosen the clustering-based approach due to it is the most suitable for ad-hoc sensor networks.

Following are explained the features and properties of the clustering-based routing algorithms that have been selected for their implementation on this work. These implemented routing protocols are LEACH, LEACH-C, the solar-aware extensions of both, HEED and a protocol based on direct transmission just as a reference in the comparison among the rest of them.

On the other hand, all these routing protocols have been implemented and simulated using the OMNeT++ 4.0, which is a freeware discrete simulation environment.

Subsequently, all the protocols have been simulated with different parameters and conditions to prove their functionality and to find out their behaviour in different sorts of sensor networks. Next, the simulations of the algorithms are compared among each other especially in terms of communication and energy efficiency. There are presented different comparisons such as LEACH and LEACH-C with their respective solar-aware extensions of both, a comparison between HEED with optimized parameters and non-optimized parameters, and finally a general comparison among One-hop, LEACH, LEACH-C and HEED.

To sum up, some conclusions are drawn about the performance of the different protocols and some key points are given for future work. Furthermore, it is presented a brief study of the environmental impact this work may have.

Master Thesis - Task Formulation

Student:	Agreda Ninot, Carlos
Supervising Professor:	Prof. Dr.-Ing. Ulrich Heinkel Chair for Circuit and System Design Chemnitz University of Technology
Supervising Assistants:	Dipl.-Ing. André Froß Dipl.-Ing. Christian Roßberg
Amount of work:	6 months

Title: Initialization algorithms for wireless ad-hoc networks

The demand of wireless sensor networks has strongly increased in recent years. Both in industrial and home applications more and more wireless sensors and actuators are utilized. In most cases no pre-existing infrastructure is available after the network activation. Hence ad-hoc capabilities are required from the wireless network nodes, which means that the network has to establish and organize its operation autonomously. During this phase of initialization and exploration all existing network nodes have to be identified and set up for the targeted network application. Limited node resources as well as a shared access to the communication medium necessitate efficient communication and routing strategies.

Aim of the master thesis is the implementation of simulation models and the simulation of energy efficient network initialization algorithms. The following items should be addressed:

- Survey of state-of-the-art strategies for network initialization and exploration in wireless ad-hoc networks. The algorithms should be compared among each other especially in terms of communication and energy efficiency.
- Implementation of simulation models of selected algorithms in the target language of a provided network simulator (OMNeT++).
- Proof of algorithmic functionality of the network initialization via network simulation.

Table of Contents

Table of Figures

Table of Tables

Introduction

Over the last recent years wireless communications has become of such fundamental importance that a world without it is no longer imaginable for many of us. Beyond the established technologies such as mobile phones and WLAN, new approaches to wireless communication are emerging being the ad-hoc and sensor networks one of the most notorious and interesting for their potential applications.

Ad-hoc sensor networks consist of a set of autonomous nodes communicating via radio without any additional backbone infrastructure. This fact is possible since the devices themselves provide the communication's infrastructure. The communication between two nodes of the network is carried out either directly between them or through intermediate nodes relaying their message in case that both are not within mutual transmission range.

On the other hand, the continued advances in microsensor technology have resulted in the development and deployment of small low cost and low power sensing devices with computational "sensing" and communication capabilities. These advances make economically possible the deployment of large numbers of nodes to form a WSN that monitors a specified parameter. Even though, sensor nodes are not very accurate and reliable individually, their deployment in large number enhances their accuracy and reliability.

The interest in the research and development of WSNs is due to their numerous advantages in front of other wireless technologies. They are easier, faster and cheaper to deploy than wired networks or other forms of wireless networks. They have a large coverage area and longer range. In addition, they have higher degree of fault tolerance than other wireless networks since a failure of one or few nodes does not affect the operation of the network. Another feature of these networks is that they are mostly unattended to, and finally, they are self-configuring or self-organizing.

In view of the great potential of ad-hoc sensor networks in a variety of application scenarios such as disaster relief, community mesh networks, monitoring and surveillance, or data gathering, it is not surprising that there has recently been a flurry of research activity in the field.

Harnessing WSN potential will provide efficient and cost-effective solution to many problems. It will require the use of new wireless sensor techniques that make these networks practical and efficient and also take into consideration sensors' limitations.

In the first chapter is presented what is a wireless sensor network and a survey of state-of-the-art strategies for network initialization and exploration in wireless ad-hoc networks.

In the second chapter are explained the properties and features of the current clustering-based routing protocols, which is the selected approach for the protocols implemented in this work.

In the third chapter is briefly presented the OMNeT++ simulator, which is the application used to implement and simulate the studied protocols.

In the fourth chapter are presented the features and the operation of the different routing protocols implemented in this work.

In the fifth chapter are explained the peculiarities of all the protocols implemented.

In the sixth chapter is presented the evaluation of the results obtained in the simulations.

Finally, in the seventh chapter is presented an environmental study, some conclusions are drawn and also some comments about future work are done.

1. Wireless Sensor Networks

A Wireless Sensor Network or WSN is supposed to be made up of a large number of sensors and at least one base station. The sensors are autonomous small devices with several constraints like the battery power, computation capacity, communication range and memory. They also are supplied with transceivers to gather information from its environment and pass it on up to a certain base station, where the measured parameters can be stored and available for the end user.

In most cases, the sensors forming these networks are deployed randomly and left unattended to and are expected to perform their mission properly and efficiently. As a result of this random deployment, the WSN has usually varying degrees of node density along its area.

Sensor networks are also energy constrained since the individual sensors, which the network is formed with, are extremely energy-constrained as well. The communication devices on these sensors are small and have limited power and range.

Both the probably difference of node density among some regions of the network and the energy constraint of the sensor nodes cause nodes slowly die making the network less dense. Also it is quite common to deploy WSNs in harsh environment, what makes many sensors inoperable or faulty. For that reason, these networks need to be fault-tolerant so that the need for maintenance is minimized.

Typically the network topology is continuously and dynamically changing, and it is actually not a desired solution to replenish it by infusing new sensors instead the depleted ones. A real and appropriate solution for this problem is to implement routing protocols that perform efficiently and utilizing the less amount of energy as possible for the communication among nodes.

Sensor devices in WSNs monitor the same event and report on them to the base station. Therefore, one good approach is to consider that sensors located in the same region of the network will transmit similar values of the attributes. This fact notices inherent redundancy in the node transmissions that may be used by the routing protocol.

Sensor networks need protocols, which are specific, data centric, capable of aggregating data and optimizing energy consumption (see [1] for more details). An ideal sensor network should have the following additional features:

- *Attribute based addressing* is typically employed in sensor networks. The attribute-based addresses are composed of a series of attribute-value pairs, which specify certain physical parameters to be sensed.
- *Location awareness* is another important issue. Since most data collection is based on location, it is desirable that the nodes know their position whenever needed.

1.1. Classification of Wireless Sensor Networks

In this subsection is presented a simple classification of sensor networks based on their mode of functioning and the type of target application.

Proactive Networks

The nodes in this sort of network periodically switch on their sensors and transmitters, sense the environment and transmit the data of interest. Hence, they provide a snapshot of the relevant parameters at regular intervals. They are well suited for applications requiring periodic data monitoring. Some known instances of this kind are the LEACH protocol [2], some improvements on LEACH such as [11,12] and PEGASIS [7].

Reactive Networks

The nodes of the networks according to this scheme react immediately to sudden and drastic changes in the value of a sensed attribute. They are well suited for time critical applications. Typical instances of this sort of networks are [1,3].

Hybrid Networks

The nodes in such a network not only react to time-critical situations, but also give an overall picture of the network at periodic intervals in a very energy efficient manner. Such a network enables the user to request past, present and future data from the network in the form of historical, one-time and persistent queries respectively. Some instances of this kind of networks are [4,5,6].

1.2. Routing Protocols

The underlying objective of any routing protocol is to render the network useful and efficient. A routing protocol coordinates the activities of individual nodes in the network to achieve global goals and do so in an efficient manner.

In the next subsection existing routing models are discussed.

1.2.1. Sorts of Routing Models

All known routing protocols may be included into one of the following three models. This classification will facilitate the analysis of the protocols that have been taken into account in this work.

One-hop model

This is the simplest approach and represents direct communication as is shown in Figure 1.1. In these networks every node transmits to the base station directly. This communication implies not only to be too expensive in terms of energy consumption, but it is also infeasible because nodes have limited transmission range. Most of the nodes in networks with large area coverage usually are far enough thus their transmissions cannot reach the base station. Direct communication is not a feasible model for routing in WSN.

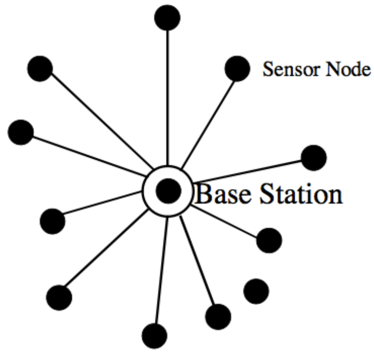


Figure 1.1. One-hop Model. (1)

Multi-hop Planar Model

In this model, a node transmits to the base station by forwarding its data to one of its neighbours, which is closer to the base station. The latter passes on it to a neighbour that is even closer to the base station as is denoted in Figure 1.2. Thereby the information travels from source to destination by hop from one node to another until it reaches the destination. Regarding to the energy and transmission range node limitations, this model is a viable approach. A number of protocols employ this approach like [7-10], and some use other optimization techniques to enhance the efficiency of this model. One of these techniques is data aggregation used in all clustering-based routing protocol, for instance in [1] and [4]. Even though these optimization techniques improve the performance of this model, it is still a planar model.

In a network composed by thousands of sensors, this model will exhibit high data dissemination latency due to the long time needed by the node information to arrive to the base station.

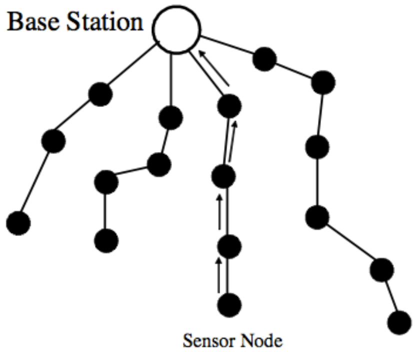


Figure 1.2. Multi-hop Model. (1)

(1) Image source: Cluster-Based Routing in Wireless Sensor Networks: Issues and Challenges [19]

Clustering-based Hierarchical Model

A hierarchical approach for the network topology breaks the network into several areas called clusters as shown in Figure 1.3. Nodes are grouped depending on some parameter into clusters with a cluster head, which has the responsibility of routing the data from the cluster to other cluster heads or base stations. Data travels from a lower clustered layer to a higher one.

Data still hops from one node to another, but since it hops from one layer to another it covers larger distances and moves the data faster to the base station than in the multi-hop model.

The latency in this model is theoretically much less than in the multi-hop model. Clustering provides inherent optimization capabilities at the cluster heads, what results in a more efficient and well structured network topology. This model is more suitable than the one-hop or multi-hop model.

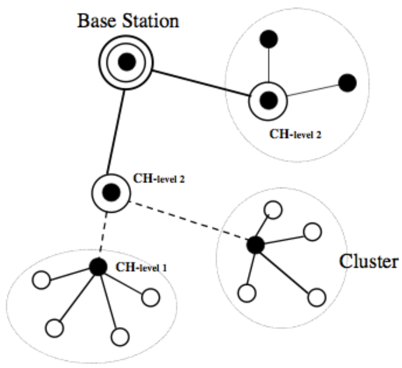


Figure 1.3. Hierarchical Clustering-based Model. (1)

The remainder of this subsection deals with the characteristics and challenges of this model and the suitability of all sorts of model.

For several reasons direct communication is infeasible for a large sensor network that is formed by thousands of sensors. It is a model that wastes energy and even worse, nodes far from base station do not have enough transmission power to reach the base station what would turn into unreachable the most part of the network. Even though the sensors would be close to the base station, the density of it would create such number of collisions that would seriously degrade the network efficiency.

The multi-hop model is a more practical approach than the one-hop. In this case, data is forwarded by hops from one node to another until it reaches the base station. Taking into account the energy constraint nodes that comprise sensor networks, it is

(1) Image source: Cluster-Based Routing in Wireless Sensor Networks: Issues and Challenges [19]

a feasible approach. The coverage area is improved over the one-hop model since most nodes are able to connect the network and the amount of collisions is reduced.

Some drawbacks of this model are the high latency in networks comprised of thousands of sensors and the serious delay that data experiences. Perhaps the most important drawback is that the closest nodes to the base station would have to act as intermediaries to all traffic being sent to the base station by the rest of the network. As they have to handle all the traffic, they will die first creating a black hole around the base station for incoming traffic. This situation will appear another time with the new closest nodes to the base station causing in the mid-term that no data arrives to the base station and rendering the network useless.

In the clustering-based hierarchical model, data is aggregated in the cluster and sent to a higher-level cluster head, thus travelling greater distances than in both other models explained and reducing time and latency. LEACH and LEACH-C use a one level clustering (see [2] and [11] respectively), whereas Chain-based 3 level PEGASIS uses three-level approach [7]. Some advantages of this model comparing with multi-hop are that data moves faster to the base station thus reducing latency and that only cluster heads perform data aggregation unlike multi-hop model where every intermediate node perform this optimization technique. Therefore, the clustering-based model is more suitable for time-critical applications than the multi-hop model.

Nevertheless, this model has one drawback since as the distance between clustering level increases, the spent energy is proportional to the square of the distance. This fact increases energy expenditure. Despite this drawback, this model outperforms by far the one-hop and multi-hop models offering a better approach to routing for sensor networks.

2. Clustering-based protocols

Grouping nodes into clusters has become into an interesting issue for the research community in order to achieve the network scalability objective. In the last years, a number of clustering algorithms have been specifically designed for WSNs [1-2], [4-6] and [11].

These techniques widely vary depending on the node deployment, the bootstrapping schemes, the network architecture, the characteristics of the cluster head nodes and the network operation model. A cluster head may also be one of the nodes or one specifically richer in resources. The overall number of cluster heads within the network and the amount of nodes per cluster may be variable or fixed by the user. Cluster heads may form a second tier network, i.e. making another level of hierarchy or they may just pass on the data to the base station.

Clustering has numerous advantages such as supporting network scalability or reducing the size of the routing table stored at each individual node. Also it allows conserving communication bandwidth since it limits the scope of inter-cluster interactions to cluster heads thus avoiding redundancy in message exchange among sensor nodes. Furthermore, clustering isolate sensor nodes of changes at the level of inter-cluster heads tier reducing topology maintenance overhead. Also, the cluster head can implement optimized techniques to enhance network operation and extend the battery life of sensor nodes. In the same way, cluster heads can schedule the cluster activity so that nodes can switch to the low-power sleep mode most of the time thus reducing power consumption. Some techniques like data aggregation reduce data redundancy in clusters thus further reducing power consumption in sensor nodes.

2.1. Clustering properties

Clustering algorithms for WSNs can be generally classified according to a set of common properties and attributes shown in the literature published so far (see [13] for more details).

2.1.1. Network model

In this subsection, it is presented a list of some relevant architectural parameters and their implications in network clustering are discussed.

Network dynamics

WSNs mainly consist of three components: sensor nodes, base stations and monitored events. Sometimes is assumed as necessary the mobility of base stations, cluster heads or even nodes, making clustering very challenging in the latter case. On the other hand, the events monitored by a sensor can be either intermittent or continual depending on the application. The former case allows the network to work in a reactive mode, simply generating traffic when reporting, for instance, forest monitoring for early fire prevention. In most cases continual events require periodic reporting thus generating significant traffic to be routed to the sink in a target tracking application, for instance.

In-network data processing

Since sensor nodes close among them might generate significant redundant data, similar packets from multiple nodes can be aggregated to reduce the number of transmissions as well as energy consumption. Data aggregation combines data from different sources by using some functions performed either partially or fully in each sensor node or in other approaches, only in more powerful and specialized nodes.

Node deployment and capabilities

The topological deployment of nodes is application dependent and affects the need and objective of the network clustering. The deployment is either deterministic or self-organizing. In deterministic situations, the sensors are manually placed and data is routed through pre-determined paths hence making clustering preset or unnecessary. However in self-organizing systems, the sensor nodes are scattered randomly creating an infrastructure in an ad-hoc manner [2,5,11], for instance.

In addition, in some setups different functionalities can be associated with the deployed nodes constraining the cluster head selection. In networks of homogenous sensor nodes, cluster heads are picked from the deployed sensors [2,7,11]. Since the sensor's communication range is usually limited and a cluster head may not be able to reach the base station, inter-cluster head connectivity becomes an important factor. On the other hand, heterogeneous WSNs may impose more constraints on the clustering process since some nodes may be designated for concrete tasks or empowered with special capabilities.

2.1.2. Clustering objectives

There are different objectives of the clustering algorithms published so far. The clustering objective is often established to fulfil applications requirements such as low data latency or data location awareness.

Next popular objectives are discussed.

Load balancing

The distribution of sensors among clusters in an evenly manner is a common goal where cluster heads perform data processing or a significant amount of tasks. Load balancing is a more pressing issue in WSNs where cluster heads are chosen from available sensor nodes [5], since it becomes crucial to avoid the exhaustion of cluster heads prematurely.

Fault-tolerance

Tolerating the failure of cluster heads is usually necessary in applications where WSN are operating in harsh environments in order to avoid the loss of important sensor's data. Assigning backup cluster heads is the most notable scheme pursued in the literature for recovery from a cluster head failure. Rotating the role of cluster heads among nodes in the cluster can also be a means for fault-tolerance besides to their load balancing advantage.

Increased connectivity and reduced delay

In WSNs, which the cluster heads are picked from sensor nodes, limiting the range of connectivity and enhancing inter-cluster heads connectivity may be more suitable than long-haul connections. On the other hand, when data latency is a concern, intra-cluster connectivity becomes a design objective or constraint.

Minimal cluster count

This is a common objective when cluster heads are specialized resource-rich nodes. In these cases their deployment is more difficult or tend to be more expensive and vulnerable than sensors.

Maximal network longevity

Since sensor nodes are constraint, the network's lifetime is a major concern especially for applications of WSNs placed in harsh environments. Adaptive clustering is a viable choice in order to achieve more network longevity.

2.1.3. Clustering attributes

In this subsection, a set of attributes is enumerated, which differentiate clustering algorithms of WSNs.

Cluster properties

It is often that clustering schemes get to achieve some characteristics for the generated clusters. The following are the relevant attributes:

- *Cluster count:* the set of cluster heads may be predetermined and thus the number of clusters is preset. If the cluster heads are picked randomly from the deployed sensors usually yields variable number of clusters.
- *Stability:* when the cluster count varies and the node's membership evolves overtime, the clustering scheme is said to be adaptive. Otherwise, it is considered fixed if sensors do not switch among clusters and the number of clusters keeps steady throughout the network lifespan.
- *Intra-cluster topology:* Some clustering schemes are based on direct communication between sensor and its designated cluster head. Nevertheless, sometimes is required multi-hop connectivity between sensor and cluster head, especially when sensor's communication range is limited or the cluster head count is bounded.
- *Inter-cluster head connectivity:* When the cluster head does not have long haul communication capabilities, cluster head connectivity to the base station has to be provisioned. Another approach is to assume that cluster head would be able to directly reach the base station.

Cluster head capabilities

As said in the previous subsections network model influences the node capabilities and the scope of the in-network processing as well as the clustering approach in general. Next, a set of attributes of the cluster head nodes is listed:

- *Mobility:* When a cluster head is mobile, the cluster would need to be continuously maintained since sensor's membership dynamically changes. Otherwise, stationary cluster heads tend to yield stable clusters and facilitate intra- and inter-cluster network management.
- *Node types:* In some setups a subset of the deployed sensors are designated as cluster heads whereas in others, cluster heads are equipped with significantly more resources.
- *Role:* A cluster head can simply forward the traffic generated by the sensors in its cluster or perform aggregation or fusion of collected sensors' data. A cluster head may act sometimes as a sink or a base station that takes actions based on the targets.

Clustering process

The coordination of the entire clustering process and the characteristics of the algorithms vary significantly among published clustering schemes. The following attributes are considered as relevant:

- *Methodology:* When cluster heads are just regular sensors nodes, clustering has to be performed in a distributed manner. In few approaches, a centralized authority partitions the nodes offline and controls the cluster membership [11]. In hybrid schemes, inter-cluster heads coordination is performed in a distributed manner, while each individual cluster head takes charge of forming its own cluster. The latter approach is common when cluster heads are rich in resources.
- *Objective of node grouping:* There are several objectives pursued for forming clusters such as fault-tolerance, load balancing, network connectivity, etc.
- *Cluster head selection:* Cluster heads can be pre-assigned or picked randomly from the deployed set of nodes.
- *Algorithm complexity:* The complexity and convergence rate of these algorithms can be constant or dependent on the number of cluster heads and sensors.

It is important to note that some of the enumerated attributes are mutually exclusive, like preset or variable cluster count, and some are not.

3. OMNeT++

OMNeT++ is a discrete event simulation environment. Its primary application area is the simulation of communication networks, but due to its generic and flexible architecture it is also used in other areas like the simulation of complex IT systems, queueing networks or hardware architectures as well.

This tool represents a framework approach. Instead of containing explicit and hardwired support for computer networks of other areas, it provides an infrastructure for writing such simulations.

OMNeT++ provides a component-architecture for models. Components *(modules)* are programmed in C++, and then assembled into larger components and models using a high-level language *(NED)*. It also has extensive GUI support and due to its modular architecture the simulation kernel and models can be embedded easily into user applications.

One important feature is that OMNeT++ is cross-platform, i.e. it is available for both operative systems based on UNIX and Windows, and it is distributed under Academic Public License. Simulcraft Inc. is currently developing the commercial version, which is called OMNEST.

Specific application areas are catered by various simulation models and frameworks, most of them open source. These models are developed completely independently of OMNeT++ and follow their own release cycles.

Some well-known OMNeT++ simulation frameworks are:

- *INET Framework:* for wired and wireless TCP/IP based simulations
- *Mobility Framework:* for mobile and wireless simulations

In the case of this work, it has been used the INET Framework. For further information about OMNeT++, see [14].

4. Protocols implemented

After having finished a survey of the state-of-the-art it was necessary to select the protocols that would be implemented.

Firstly, LEACH, i.e. LEACH-distributed, was selected due to the fact that is the first well known clustering-based routing protocol and all the subsequent clustering-based protocols are based on it or are referred to it somehow. Therefore, it was a good first step to start with.

Other interesting protocols that were selected to be implemented were LEACH-C, i.e. LEACH-centralized, created by the same authors of LEACH and also the solar-aware extensions of both, which were found in [14] with the original paper [15].

Finally, a more complex protocol, which is called HEED, was chosen since it is currently one of the most well known and mentioned routing protocols. Moreover, some published surveys as [13] show its suitable features and good results. The implementation of HEED is based on the pseudo-code that is provided in the original paper [5].

Therefore, different protocols were selected for their implementation and simulation. These protocols differ in their complexity, the strength and number of assumptions they make and the goals they have.

Once the programming of all these protocols was finished it was necessary to create and implement one basic protocol to compare the rest of them with it. The simplest approach for routing protocols is the One-hop that has been implemented for this work since it is a good simulation to see whether the compared protocols are energy-efficient or not and how much they elongate the batteries lifetime.

Therefore, in this work is presented a comparison among four protocols, i.e. One-hop, LEACH, LEACH-C and HEED, and two solar-aware extensions, i.e. Solar-aware LEACH and Solar-aware LEACH-C.

Following is explained the performance of the algorithms that have been selected in order to be implemented on OMNeT++.

4.1. One-hop

This protocol is the easiest and simplest routing approach and has been implemented to establish a reference for the comparison among the different protocols. It is based on the assumption that every node is able to reach the base station, otherwise it would be impossible the communication between every node and the base station.

The operation of this protocol is quite simple. In every round the base station receives a status message from all nodes, which points out to the base station the position and parameters of the node. Once the base station has received all the messages it creates a TDMA schedule telling each node when it can transmit the data and how many times this process is repeated. Once all nodes have sent all the data packets regarding to the current round, they send another status message in order to start the next round.

4.2. LEACH-distributed

LEACH-distributed or LEACH [2] is a self-organizing, adaptive clustering protocol that uses randomization to distribute the energy load evenly among the sensors in the network.

LEACH makes some assumptions about both the sender nodes and the underlying network, being some of them very strong. LEACH assumes that all sensor nodes can adapt their transmission range. Furthermore, energy consumption during transmission scales exactly with the distance and every sensor node is able to reach a base station (BS). Moreover, nodes support several MAC layers and perform signal-processing functions.

LEACH uses a distributed algorithm to determine the cluster heads in the set-up phase whereas in the steady phase nodes send their data according to the time schedule provided by their cluster heads. This operation of LEACH is divided into rounds as shown in figure 4.1.

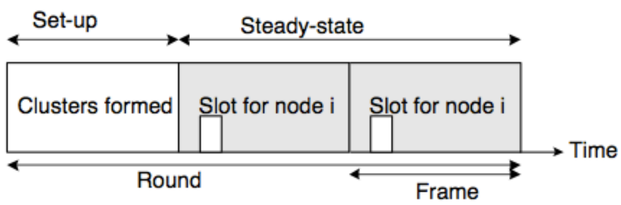


Figure 4.1. LEACH operations (1)

Advertisement Phase

When clusters are created, each node *n* autonomously decides if it will be a cluster head for the next round. The selection is stochastic and each node determines a random number between 0 and 1. If this number is lower than a threshold *T(n)*, the node becomes a cluster head. *T(n)* is determined according to the equation

$$T_1(n) = \frac{P}{1 - P * (r \bmod \frac{1}{P})}, \quad \textbf{(1)}$$

for nodes that have not been cluster head in the last 1/*P* rounds, otherwise $T_1(n)$ is zero. Here *P* is the desired percentage of cluster heads and *r* is the current round. Using this algorithm, each node will be a cluster head exactly once within 1/*P* rounds. After 1/*P* – 1 rounds, $T_1(n)$ = 1 for all nodes that have not been a cluster head. When a node has elected itself as a cluster head, it broadcasts an advertisement message

(1) Image source: Solar-aware Clustering in Wireless Sensor Networks [15]

telling all nodes that it is a cluster head. This advertisement is done using a CSMA MAC protocol. Non-cluster heads use these messages from the cluster heads to choose the cluster they want to belong for this round based on the received signal strength of the advertisement message.

Cluster Set-Up Phase

After each node has decided to which cluster it belongs, it must inform the cluster head node that it will be a member of its cluster. Each node transmits this information back to the cluster head again using CSMA MAC protocol. During this phase, all cluster head nodes must keep their receivers on.

Schedule Creation

The cluster head receives all the messages from the nodes that would like to join the cluster. Based on the number of nodes in the cluster, the cluster head creates a TDMA schedule telling each node when it can transmit the data. This schedule is broadcasted back to the nodes included in the cluster.

Data Transmission

Once the clusters are created and the TDMA schedule is fixed, nodes can start to transmit their data. Assuming nodes always have data to send, they send it during their allocated transmission time to the cluster head. This transmission uses the minimal amount of energy based on the received strength of the cluster head advertisement. The radio of each non-cluster head can be turned off until the node’s allocated transmission time, thus minimizing energy dissipation. The cluster head node must keep its receiver on to receive all the data from the nodes in the cluster. Once all the data has been received, the cluster head performs optimization functions such as data aggregation or other signal processing functions to compress the data into a single signal. This composite signal, which is a high-energy transmission since the base station is far away, is then sent to the base station. The cluster heads send these data packets using a fixed spreading code with CSMA.

This is the steady-state operation of LEACH networks. After a certain time, which is determined a priori, the next round begins with each node determining if it will become a cluster head for this round and advertising the decision to the rest of nodes as described in the advertisement phase.

4.3. Solar-aware LEACH-distributed extension

A solar-aware version of the LEACH-distributed protocol [15] should preferably choose solar-driven nodes as cluster heads. In order to get this end, Equation 1 must be modified in at least two ways:

- Solar-powered nodes must become cluster heads with a higher probability
- A node that has been solar-powered while being a cluster head should be able to become a cluster head again even during the next 1/*P* rounds.

The first condition can easily be achieved by adding a factor in the right side of Equation 1.

To accomplish the second condition, Equation 1 has to be reformulated to incorporate the number of cluster heads since the start of the last *metaround*, where *metaround* corresponds to the 1/*P* rounds. This number, which is called *cHeads* increases every round, in average by *P* multiplied by the number of nodes. When *cHeads* equals the number of nodes (*numNodes*), the current *metaround* is finished and *cHeads* is reset to zero. Note that whereas in LEACH-distributed each *metaround* consists of exactly 1/*P* rounds, in the solar-aware extension one *metaround* will now consist of 1/*P* rounds on average. The reformulation of Equation 1 is therefore

$$T_2(n) = sf(n) * \frac{P}{1 - (\frac{cHeads}{numNodes})}, \tag{2}$$

for all nodes, except for those that have been cluster head during the current *metaround* while they were battery-driven. For those nodes $T_2(n)$ is zero, i.e. they can not become cluster heads again during the current *metaround*. Since the cluster heads announce themselves, every node knows the value *cHeads*. The equation contains a scaling factor $sf(n)$, which is >1 for solar-powered nodes and set to the reciprocal for battery-powered nodes.

4.4. LEACH-centralized

Though there are some advantages using LEACH-distributed cluster formation algorithm, this protocol offers no guarantee about the placement and/or number of cluster heads. LEACH-centralized (LEACH-C) [11], is a protocol that uses a centralized clustering algorithm and the same steady-state protocol as LEACH. This method to form the clusters may produce better clusters by dispersing the cluster head nodes throughout the network.

During the set-up phase of LEACH-C, each node sends information about its current location (possibly determined using a GPS receiver) and energy level to the BS. The BS needs to ensure that energy load is evenly distributed among all the nodes besides of determining good clusters. With this aim, the BS computes the average node energy and whichever nodes have energy below this average can not be cluster heads for the current round.

Using the remaining nodes as possible cluster heads, the BS finds clusters using the simulated annealing algorithm [16] to solve the NP-hard problem of finding *k* optimal clusters [17]. This algorithm attempts to minimize the amount of energy for the non-cluster head nodes to transmit their data to the cluster head, by minimizing the total sum of squared distances between all the non-cluster head nodes and the closest cluster head.

Since the authors of LEACH do not present the detailed algorithm the BS uses to choose *k* cluster heads, it is used simple heuristics. The process consists of three steps: In step 1, the *k+3* nodes with the highest remaining energy are selected. In step 2, the potential cluster head with the minimal sum of the distances to all other potential cluster heads is removed. In step 3, it is removed one of the two potential cluster heads that have the closest distance to each other. If one of these two nodes is close to the border of the sensor area network, this node is removed. Otherwise the node closer to the centre of the sensor area network is removed. When removing the third node, the total sum of the square distance between non-cluster heads and their potential cluster head is minimized.

Once the cluster heads and associated clusters are found, the BS broadcasts a message that contains the cluster head ID for each node. If a node's cluster head ID matches its own ID, the node is a cluster head; otherwise, the node determines its TDMA slot for data transmission and goes to sleep until it is time to transmit data. The steady-state phase of LEACH-C is identical to that of LEACH.

4.5. Solar-aware LEACH-centralized extension

The aim of Solar-aware LEACH-C (sLEACH-C) [15] is to extend the lifetime of the sensor network by preferably choosing solar-powered nodes to perform the energy intensive task of being a cluster head.

In sLEACH-C besides the remaining energy and the position, nodes also transmit their solar status to the BS. In this solar-aware extension, the algorithm used by the BS to determine the cluster heads is slightly modified. In step 1, from the energy level of each node is subtracted *e*, which is the assumed energy consumption for the next round assuming the node was a cluster head. If a node is solar-powered, it is assumed it will remain solar-powered for half of the round and thus is subtracted only *e*/2. The *k+3* nodes with the highest energy value are chosen. In step 2, no solar-powered node is removed if it is possible. In step 3, if one of the nodes with the closest distance to each other is solar-powered, it is not removed otherwise the last step does not change.

4.6. HEED

HEED (Hybrid, Energy-Efficient, Distributed) clustering protocol considers a hybrid of energy and communication cost.

This approach only assumes that sensor nodes are able to control their transmission power level and does not make assumptions about the distribution of the nodes or their capabilities. A node only knows about the other nodes within its reachable range, which implies that nodes base their decisions only on local information. Further explanations about the requirements HEED must meet are in [5].

HEED bases cluster head selection primarily on the residual energy of each node, which can be estimated, and intra-cluster communication cost as a secondary clustering parameter. In the latter case, cost can be a function of neighbour proximity or cluster density. The transmission power level used for intra-cluster announcements and during clustering determines the cluster range or radius. This level is referred as cluster power level and it dictates the amount of clusters in the network.

Protocol operation

The clustering process is triggered repeatedly after every clustering process time and network operational interval. At each node, the clustering process requires a number of iterations called N_{iter}. Every step should be long enough to receive messages from any neighbour within the cluster range.

Also it is set an initial percentage of cluster heads among all nodes in the network called C_{prob}. It is only used to limit the initial cluster head announcements. Before executing HEED, a node sets its probability of becoming a cluster head, CH_{prob}, as follows

$$CH_{prob} = C_{prob} * \frac{E_{residual}}{E_{\max}}, \tag{3}$$

where $E_{residual}$ is the estimated current residual energy in the node and E_{max} is a reference maximum energy (corresponding to a fully charged battery), which is typically identical for all nodes.

During any iteration i, $i < N_{iter}$, every uncovered node elects to become a cluster head with probability CH_{prob}. After step i, the set of tentative cluster heads, S_{CH}, is updated and a node v_i selects its cluster head to be the node with the lowest cost in S_{CH}. Every node then doubles its CH_{prob} and goes to the next step.

If a node elects to become a cluster head, it sends an announcement message where the selection status is set to *tentative_CH*, if its CH_{prob} is less than 1, or *final_CH*, if its CH_{prob} has reached 1. A node considers itself covered if it has heard from either a *tentative_CH* or a *final_CH*. If a node completes HEED execution without selecting a cluster head that is *final_CH*, it considers itself uncovered, and announces itself to be a cluster head with state *final_CH*. A *tentative_CH* node can become a regular node at a later iteration if it finds a lower cost cluster head.

The inter-cluster communication to allow all the cluster heads to send the data they have aggregated from their cluster members is not explained, but there are some statements and requirements over it. The communication among the cluster heads to allow all the data to reach the base station should be based on a multi-hop approach.

5. Simulations of the protocols

All the models of the protocols have been simulated with OMNeT++ just as all the simulations have been done on it. Next are briefly explained the implementation of the different protocols, their features and how much differ from the original protocols.

One-hop

This protocol is the simplest approach for the communication of a wireless sensor network. The implementation has been completely done from scratch for this master thesis. The operation of the protocol is presented in section 4.1.

LEACH-distributed and Solar-aware LEACH-distributed

Both simulations presented in this work are based on the Solar-aware LEACH-distributed simulation found in [14] as creation for the paper [15] made by Thiemo Voigt et.al.

The original implementation of the protocol mentioned above had to be migrated to OMNeT++ 4.0, since it was programmed for the version 3.0. This implied to change a large number of functions and parameters as the original ones were deprecated.

First of all is necessary to follow the migration file that can be found in the OMNeT++ website [14] and afterwards the file provided by the INETMANET distribution since this simulation model is used in the implementations.

Once the migration process has finished, it is still necessary to make some other changes that include the connections between each node and the base station, the connections among all nodes or some other parameters such as the functions regarding to simulation time.

Some of the changes mentioned above can be seen in Appendix A.1.

LEACH-centralized and Solar-aware LEACH-centralized

Both simulations are based on the Solar-aware LEACH-centralized simulation found in [14] as creation for the paper [15]. As for the last simulations mentioned above, the code had to be migrated to the current new version of the simulator OMNeT++ with all the modifications of the code it implies.

Firstly, was also necessary to follow the migration files provided in [14] and the INETMANET distribution, and later to make the necessary changes of functions and parameters directly on the code.

Furthermore, the *handover* function was completely done for this work since it was not included in the original code. This function outperforms the results since it forces the solar-driven nodes to become cluster heads for the next round. This function improves energy-efficiency since solar-powered nodes consume less amount of energy than the ones that run on their battery.

Finally, some of these changes besides the *handover* functions are presented in Appendix A.2.

HEED

It has been created from scratch for this master thesis and based on the pseudo-code shown in [5]. The intra-cluster communication cost parameter, which is the secondary parameter for the cluster head election process, is based on the closest node. However, it may be pretty easy to modify this parameter in order to take into account the *average minimum reachability power* (AMRP) instead of the current one.

In order to synchronize the end of the cluster election process in all nodes it has been used a function, which sends the status of each node to all nodes that are within its cluster range. This implies a further waste of energy and instead of this, in a future implementation could be used a synchronization technique such as RBS [25] as is pointed out in [5].

The inter-cluster communication is based on a simple multi-hop strategy, which allows all cluster heads to send the data directly to the base station or to send it to a cluster head even closer to the base station, otherwise.

6. Evaluation of the results

In this section are analysed the results of the simulations done with the different protocols implemented in this work.

All the data used for the graphs and tables that are presented in this chapter have been taken out from the tables that are shown in Appendix B.

6.1. Simulation parameters

In table 6.1 are shown the parameters that have been used for the simulations of the implemented protocols.

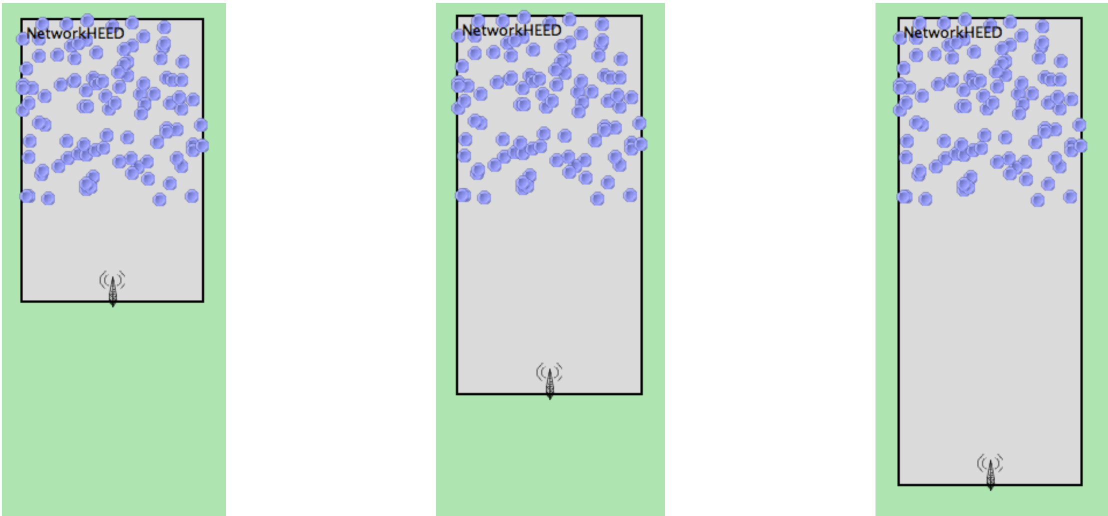
Table 6.1. Parameters used in simulations

Type	Parameter	Value
Network	Network grid	From (0,0) to (100,100) (1), to (200,200) (2) and to (300,300) (3)

	Sink (BS)	For (1), At (50,150), (50,200), (50,250)
		For (2), At (100,250), (50,200), (50,250)
		For (3), At (150,350), (150,400), (150,450)

	Initial energy	0.5 J/battery
Application	Cluster radius (only for HEED)	R_c = 25 / 100 / 150; R_t = 50 / 100 / 150 / 200 / 250
	Data packed size	100 Bytes
	Broadcast packet size	25 Bytes
	Packet header size	25 Bytes
	Round (frames)	5 / 10 / 20 TDM frames
Radio model	Eelec	50 nJ/bit
	Efs	10 pJ/bit/m^2
	Efusion	5 nJ/bit/signal

The network grid shows the different areas of network used in the simulations, which are 100x100 (m), 200x200 (m) and 300x300 (m), respectively. For each one of these areas it has been simulated a network where the base station is placed 50, 100 and 150 meters to the closest node of the network to find out which is the behaviour of the protocols when the base station is placed at different distances far from the sensor nodes. An example of the three different distances where the base station is placed in every size of area network is shown in the figure 6.1. In this figure is represented a network composed by 100 nodes with an area of 100x100 meters. In figure 6.1.a) the base station is placed at (50,150) meters, i.e. it is 50 meters far to the closest node, the figure 6.1.b) shows a base station placed at (50,200) meters, i.e. it is 100 meters far to the closest node, and finally the figure 6.1.c) depicts a base station placed at (50,250) meters.



a) b) c)

Figure 6.1. Base station positions. a) At 50 m.; b) At 100 m.; c) At 150 m.

For every one of these conditions each protocol has been simulated with rounds composed by 5, 10 or 20 TDM frames, which results in a shorter or longer steady phase (see Figure 4.1.).

Finally, for every one of the cases explained above have been carried out 5 different simulations, each of them using a different seed for the creation of random numbers. From these 5 output files has been calculated the average of the rounds done when the first node was died (network lifetime) and the rounds achieved until half of the nodes were died.

On the other hand, the cluster radius (R_c) and the inter-cluster transmission range (R_t) are parameters only used in HEED and they have been configured in the following manner:

- 100x100 (m) area network:
  - BS position at:
    - (50,150) (m): R_c = 25 (m) and R_t = 50 (m)
    - (50,200) (m): R_c = 25 (m) and R_t = 100 (m)
    - (50,250) (m): R_c = 25 (m) and R_t = 150 (m)

- 200x200 (m) area network:
  - BS position at:
    - (100,250) (m): R_c = 50 (m) and R_t = 100 (m)
    - (100,300) (m): R_c = 50 (m) and R_t = 150 (m)
    - (100,350) (m): R_c = 50 (m) and R_t = 200 (m)

- 300x300 (m) area network:
  - BS position at:
    - (150,350) (m): R_c = 75 (m) and R_t = 150 (m)
    - (150,400) (m): R_c = 75 (m) and R_t = 200 (m)
    - (150,450) (m): R_c = 75 (m) and R_t = 250 (m)

Both, R_c and R_t have been chosen taking into consideration that the transceivers of the nodes are able to switch the transmission power among 7 different power levels at least as is assumed in the original paper [5], and that these power levels fit the distances listed above.

Regarding to the initial energy parameter, a full-charged battery energy level of 0.5 Joules has been chosen since it is enough to see the differences on the results among the different protocols evaluated.

In all the simulations the network always consists of 100 nodes, hence in large area networks such as 300x300 m. the node density is lower than in small area networks like 100x100 m.

6.2. LEACH-distributed versus Solar-aware LEACH-distributed

In this subsection is shown a comparison with the results of the simulations of LEACH and its solar-aware extension.

The evaluated results as explained above are related to the number of rounds done until half of the nodes are dead or when the first node is dead, where the latter case is also called network lifetime.

6.2.1. Half-dead network

In this subsection can be seen the differences in the outcomes of both protocols.

In the case of a short steady phase, i.e. composed by 5 frames, the solar-aware extension shows a higher number of rounds achieved than the original LEACH-distributed version. However both show a similar behaviour when the base station (BS) is placed at different distances, getting worse the farther the BS is to the closest node as can be observed in figure 6.2.

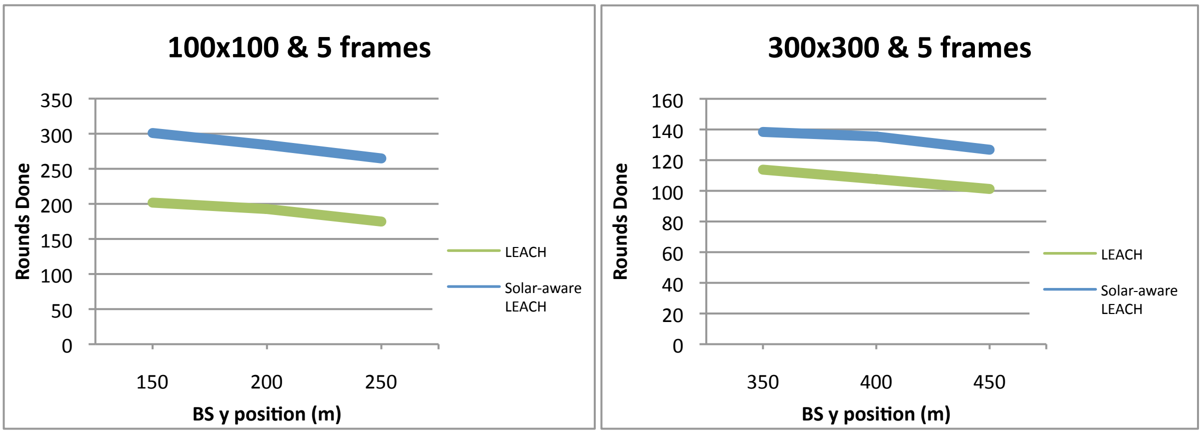


Figure 6.2. LEACH and Solar-aware LEACH results with 5 frames.

When the steady phase has the double number of frames than the previous case, i.e. the duration of the steady phase is doubled, the behaviour of both protocols remain the same but decreasing the number of rounds achieved up to almost the half of them, as is shown in figure 6.3. This situation can be explained as an example of a low-cost set-up phase in energy terms, but a high-cost steady phase due to a non-optimal election of the cluster heads and the direct communication between cluster heads and base station.

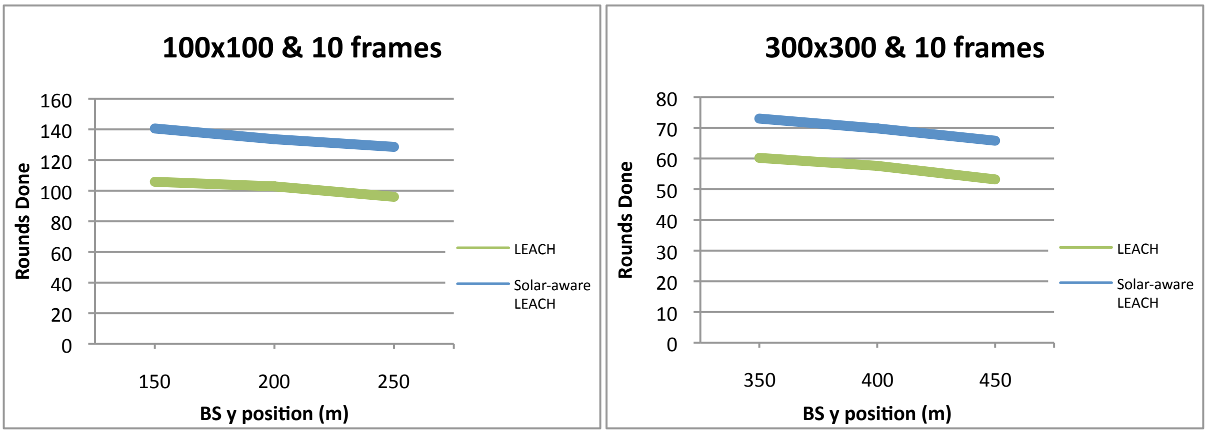


Figure 6.3. LEACH and Solar-aware LEACH results with 10 frames.

In figure 6.4 can be observed the results of both protocols with the longest steady phase simulated. The outcomes are really similar to the previous ones as expected, but decreasing the overall amount of rounds achieved.

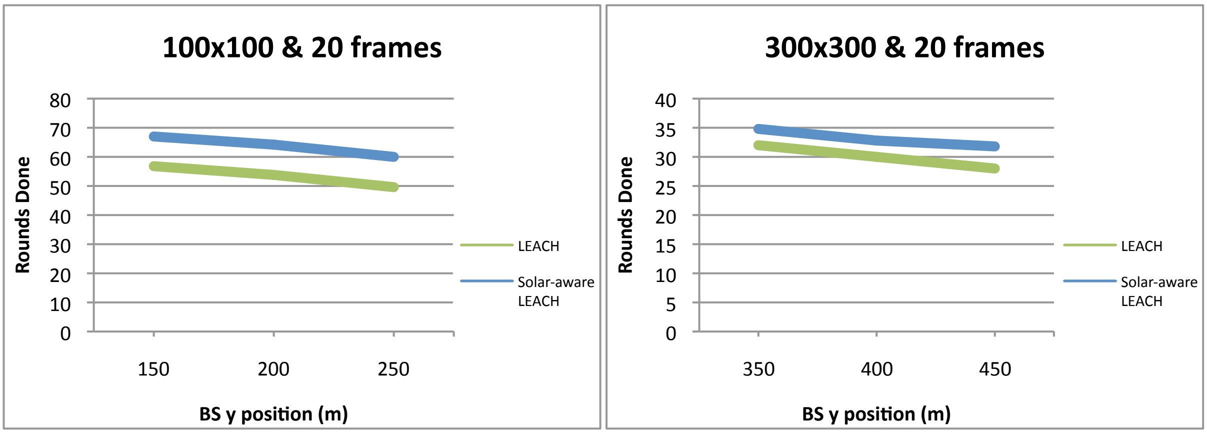


Figure 6.4. LEACH and Solar-aware LEACH results with 20 frames.

Looking at the figures 6.2, 6.3 and 6.4, can be noticed that the longer the steady phase the smaller the difference in the outcomes between both protocols is. It can be explained, as the solar-aware extension is more effective when the steady phase is short and the cluster head election is repeated in a short time. This situation is caused by the fact the election of solar-driven nodes as cluster heads happens on most cases and the duration of the solar state is usually shorter than the steady phase duration. Therefore the longer the steady phase the higher probability of a solar-driven node to turn into a battery-driven one is, what could result in a higher energy consumption in nodes that have been solar-driven more often than not within the cluster head election rounds.

6.2.2. First node dead

In the following figures can be seen the rounds achieved by both protocols when the first node dies.

In the case of a short steady phase and a small area network the solar-aware extension gets better results, which achieves even more than 2 times the lifetime of the LEACH-distributed as is shown in figure 6.4. When the node density decreases or the area network increases, the results of both protocols get closer being still better in the case of Solar-aware LEACH.

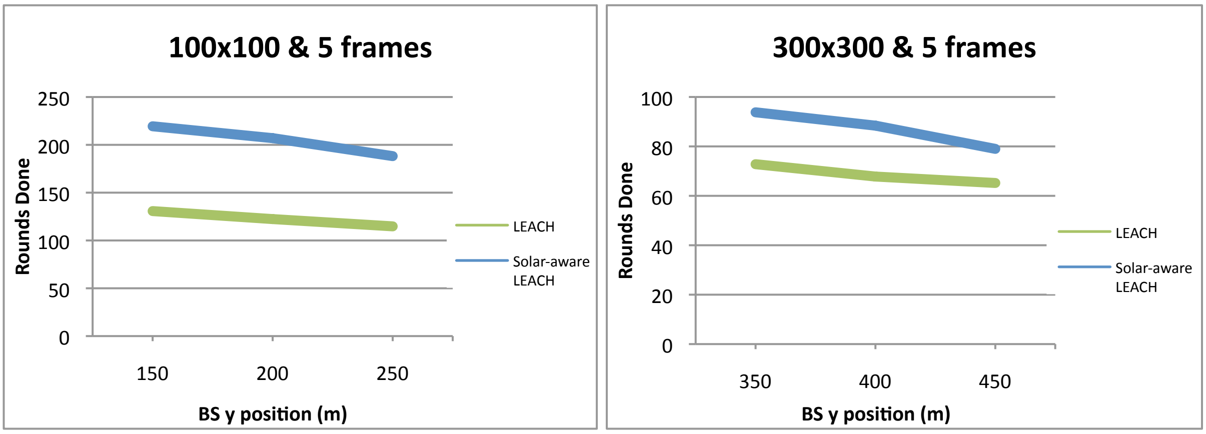


Figure 6.5. LEACH and Solar-aware LEACH results with 5 frames.

If the duration of the steady phase increases, the results of both protocols are really similar as can be observed in figure 6.6. Even though the Solar-aware LEACH still achieves a longer lifetime, the difference between them is not very noticeable in large area networks, chiefly. Both protocols get worse results the farther is the BS to the closest node.

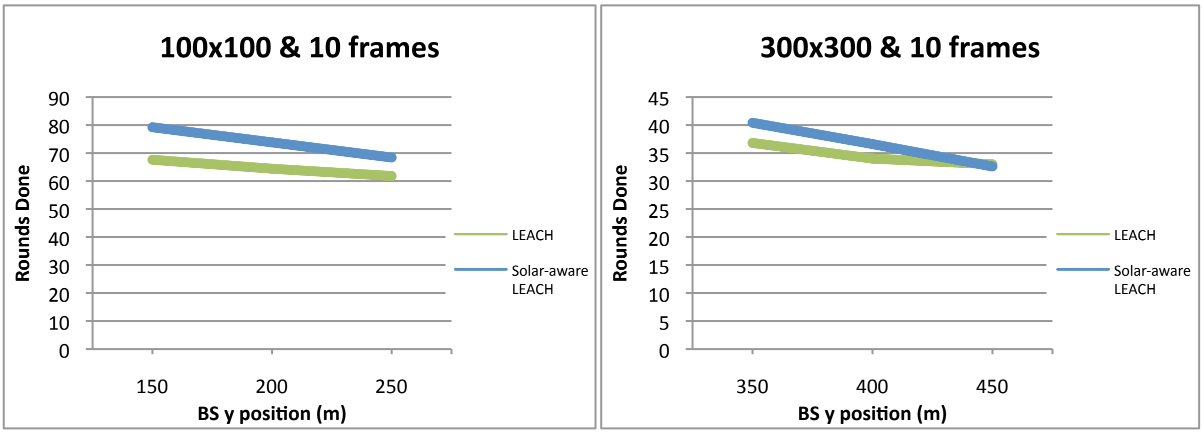


Figure 6.6. LEACH and Solar-aware LEACH results with 10 frames.

Figure 6.7 depicts the results of both protocols analysed in this section with the longest steady phase simulated. It can be seen the higher number of rounds achieved by LEACH-distributed unlike the previous cases with shorter steady phases, where the solar-aware extension gets better results. Also in this case, both protocols shorten the network lifetime the farther the BS to the closest node is.

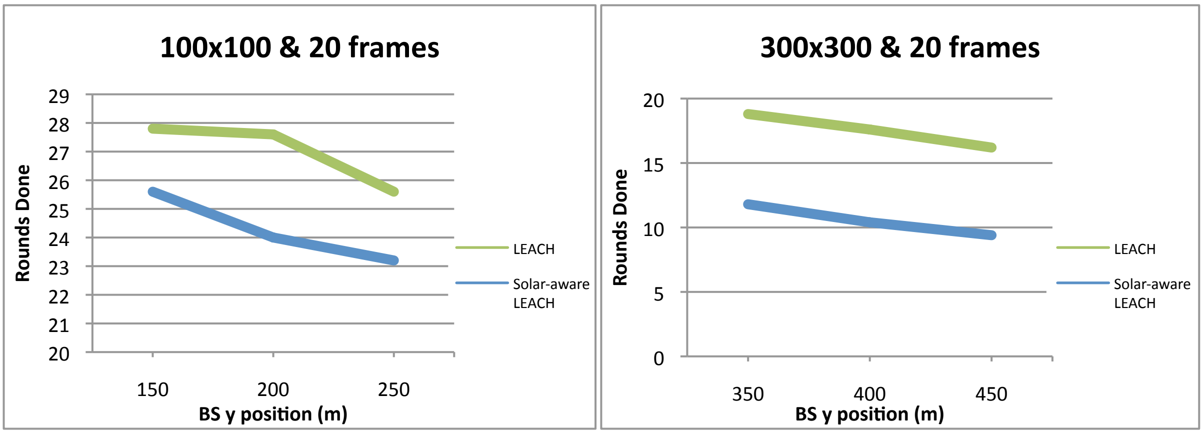


Figure 6.7. LEACH and Solar-aware LEACH results with 20 frames.

Analysing the figures 6.5, 6.6 and 6.7, it can be noticed that LEACH-distributed achieves better results the longer the steady phase is. As explained in the previous subsection, the duration of the steady phase influences drastically in the lifetime achieved by both protocols. In this case, the non-optimal election of a cluster head based only on its solar-aware status and local information causes a huge worsening in the performance of the Solar-aware LEACH with long steady phases.

6.3. LEACH-centralized versus Solar-aware LEACH-centralized

In this subsection is shown a comparison with the outcomes of the simulations of LEACH-C and its solar-aware extension.

The evaluated results as in the previous subsection are related to the number of rounds achieved when half of the nodes are already dead or when the first node dies.

6.3.1. Half-dead network

In this subsection are shown the differences in the outcomes of both protocols with different simulation parameters.

Figure 6.8 depicts the number of rounds achieved by both protocols when the steady phase is short, just 5 frames. As can be noticed Solar-aware LEACH-C performs better in all situations than the original LEACH-C.

On the other hand, results get closer to each other the larger the area network and the farther the BS is, as can be seen in figure 6.8.

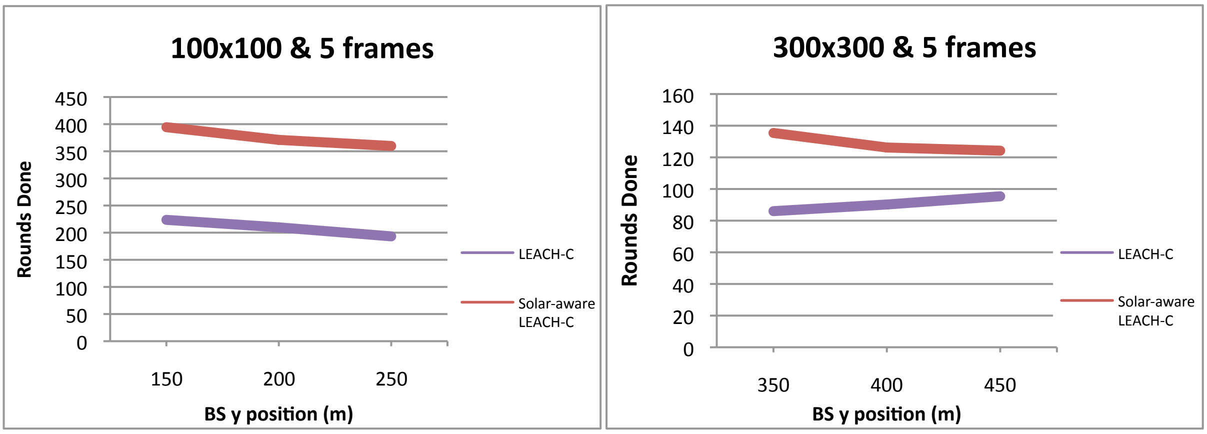


Figure 6.8. LEACH-C and Solar-aware LEACH-C results with 5 frames.

In the case of a steady phase formed by 10 frames, the results of both protocols show a similar behaviour than the previous case as shown in figure 6.9. Their respective outcomes get closer to each other in larger area networks and LEACH-C shows a more constant behaviour than the solar-aware extension with the base station placed at different distances.

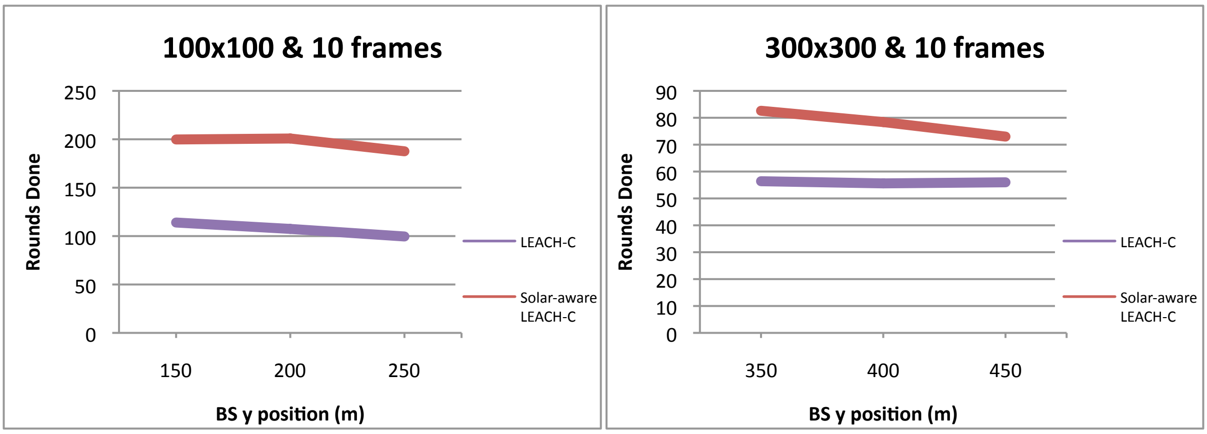


Figure 6.9. LEACH-C and Solar-aware LEACH-C results with 10 frames.

Finally, in the case with the longest steady phase simulated the results of both protocols show a similar behaviour, but a higher number of rounds achieved by the solar-aware extension as can be observed in 6.10. Both protocols get worse results, the farther the BS and the larger the area network is.

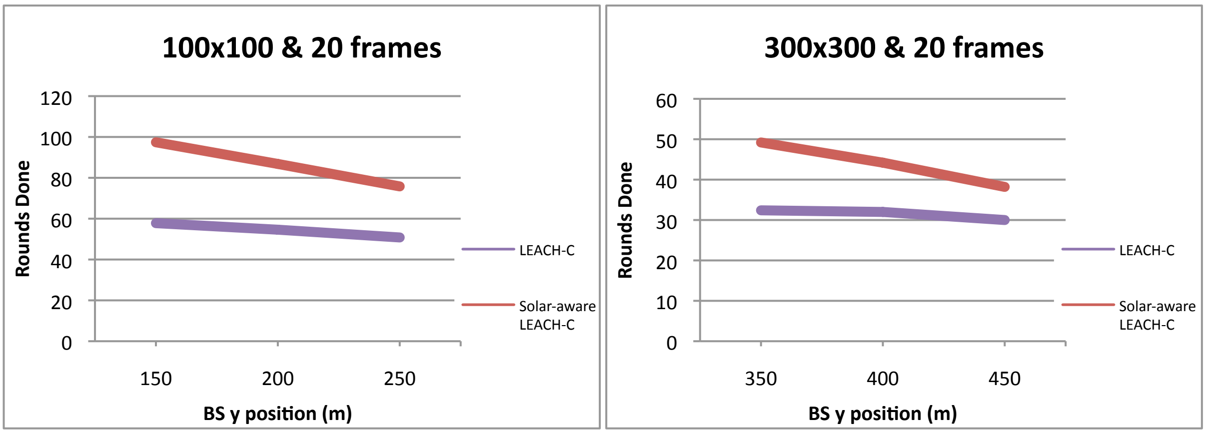


Figure 6.10. LEACH-C and Solar-aware LEACH-C results with 20 frames.

As can be easily seen analysing the figures 6.8, 6.9 and 6.10, the behaviour of both protocols when half of nodes are already dead keeps constant regardless the duration of the steady phase and the difference between both decreases the farther the base station is to the closest node.

6.3.2. First node dead

In this subsection are shown the number of rounds achieved by both protocols when the first node dies, i.e. the network lifetime.

Figure 6.11 depicts the outcomes of LEACH-C and Solar-aware LEACH-C with a short steady phase of 5 frames. In this case the solar-aware extension outperforms LEACH-C in more than 50% over the overall number of rounds when the base station is placed at the closest distance simulated and results of both protocols get closer to each other the farther the BS is.

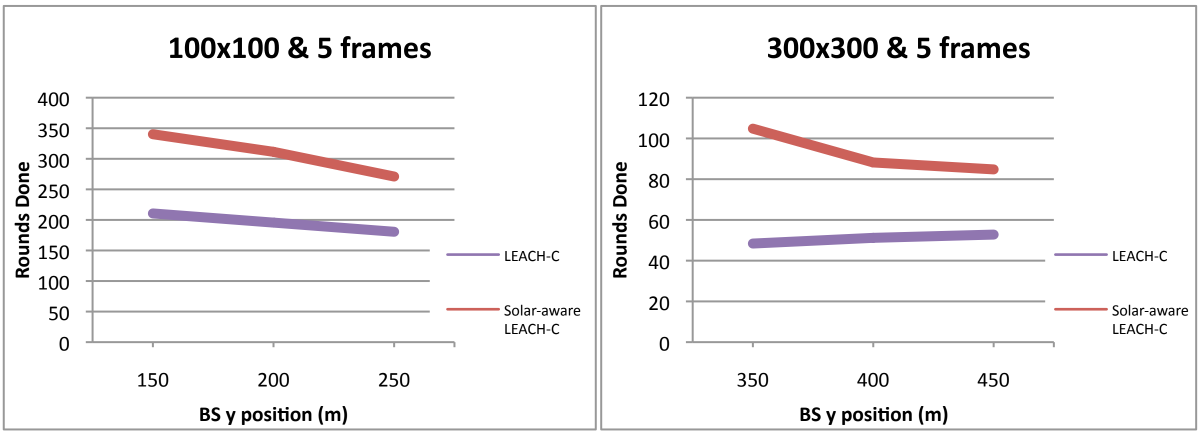


Figure 6.11. LEACH-C and Solar-aware LEACH-C results with 5 frames.

In the case of a steady phase of 10 frames, which is shown in figure 6.12, the solar-aware extension still gets higher number of rounds before the first node dies with a base station close to the node network, but it is noticeable to say that both show almost the same results with a BS far from the network regardless the area network, i.e. node density.

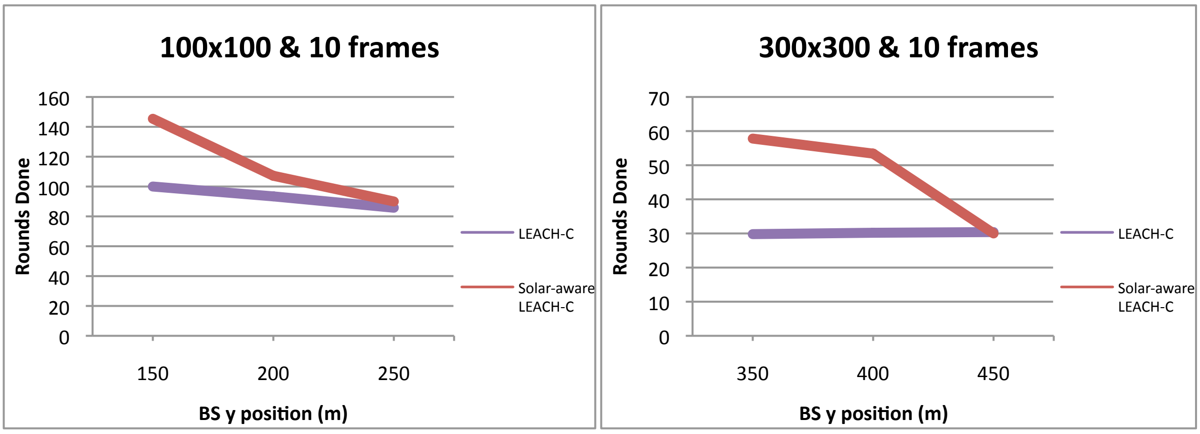


Figure 6.12. LEACH-C and Solar-aware LEACH-C results with 10 frames.

Finally, the results of both protocols obtained with a steady phase of 20 frames are shown in figure 6.13. Both protocols get close results to each other, however in cases where the base station is placed at short distances to the closest node the Solar-aware LEACH-C gets a longer lifetime.

On the other hand, LEACH-C outperforms the results of its solar-aware extension when the BS is placed far away.

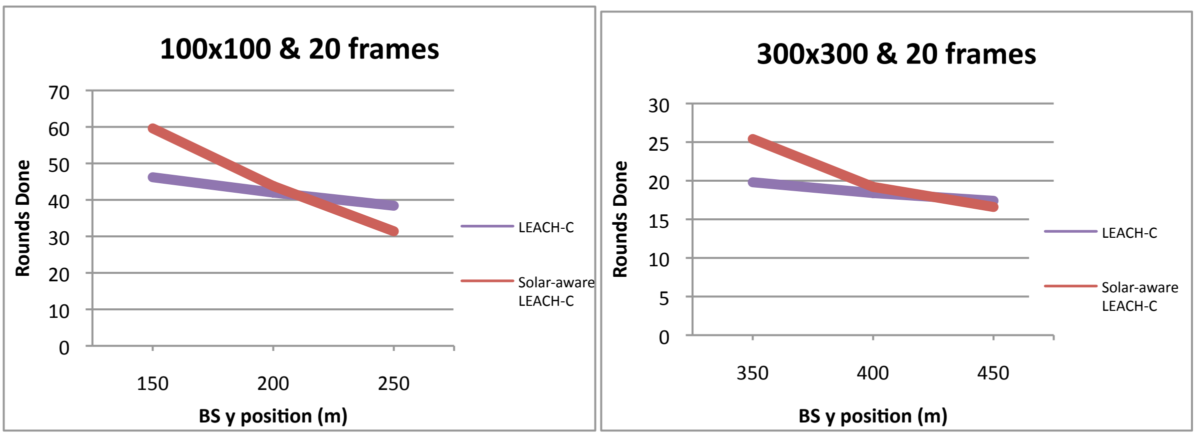


Figure 6.13. LEACH-C and Solar-aware LEACH-C results with 20 frames.

As can be observed looking at the figures 6.11 6.12 and 6.13, the results of both protocols get closer to each other the longer the steady phase is. As explained above for the case of LEACH and its solar-aware extension, the efficiency of the solar-aware extension decreases up to achieve almost the same lifetime than the original

LEACH-C. This can be justified in the same terms as the previous protocols, where the election of solar-driven nodes as cluster heads is effective when the steady phase is no much longer than the average time a node can be solar-powered. In cases of a steady phase with long duration, a cluster head considered solar-powered, which always has higher priority than the non-solar ones to become a cluster head, may stay long time performing high-cost functions as a battery-driven cluster head, thus resulting in an faster depletion of its battery.

This situation is more critical for the measurement of network lifetime than for the case of half-dead network, since it could occasionally cause a faster death of a few nodes but not the death of a great number of them such as the half of the network.

6.4. HEED

In this subsection is compared the HEED protocol with optimal cluster radius and inter-cluster communication range parameters with the HEED protocol with non-optimal parameters.

Foremost is important to state that this optimization is feasible if the nodes take into consideration the power level of the received signals to create and set up the clusters instead of simple distance metrics. Then, this approach is useful to show the real efficiency of this protocol compared to the rest of the simulated protocols shown in this work.

The cluster radius and inter-cluster communication range parameters used in the optimized HEED are mentioned in section 6.1. For the non-optimized HEED the parameters are the following:

- 100x100 (m) area network:
  - Rc = 25 (m) and Rt = 150 (m)
- 200x200 (m) area network:
  - Rc = 50 (m) and Rt = 300 (m)
- 300x300 (m) area network:
  - Rc = 75 (m) and Rt = 450 (m)

As can be noticed regarding the chosen parameters there is no difference among the distances where is placed the base station.

In the same manner as the protocols compared above, the results of both protocols are relative to each other according to the number of rounds achieved until half of the nodes are already dead or when the first node dies.

6.4.1. Half-dead network

Following are shown the differences in the outcomes between an optimized HEED protocol and a non-optimized version with different simulation parameters.

Figure 6.14 depicts the results obtained by both protocols with a short steady phase and two different area networks. As can be noticed in both cases the optimized

HEED gets better results than the non-optimized HEED besides it shows a really constant behaviour in all kind of situations. In the graph with smaller area network the non-optimized HEED get closer to the other the farther the BS is to the closest node. This situation is due to the coincidence of the cluster radius and inter-cluster communication range selected for both HEEDs. In larger area networks the difference between the inter-cluster communication ranges of both protocols increases proportionally to the area network.

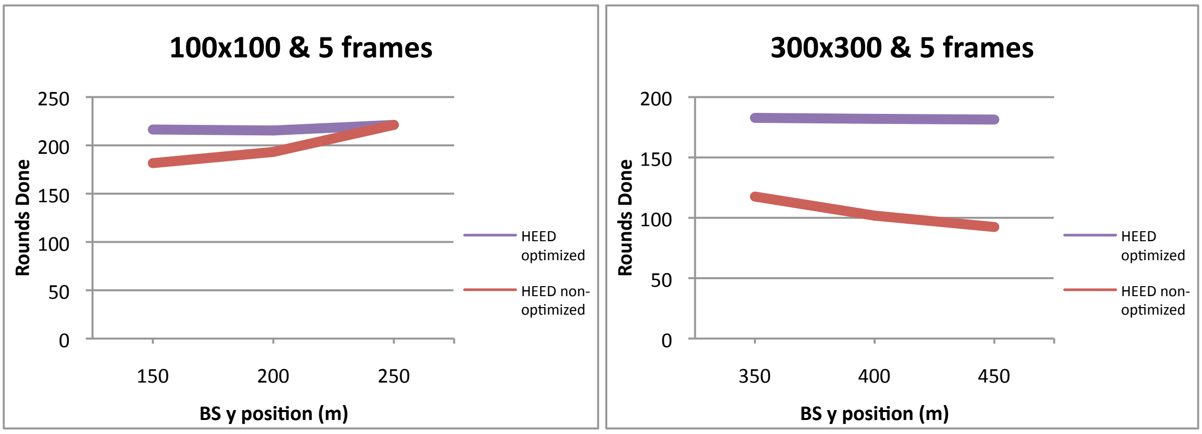


Figure 6.14. HEED optimized and HEED non-optimized results with 5 frames.

In the case of a steady phase of 10 frames the comparison between both HEED simulations shows the same behaviour as previously as can be noticed in figure 6.15. The only difference is the lower number of rounds achieved by both simulations that approximately fits the half of the overall rounds achieved before.

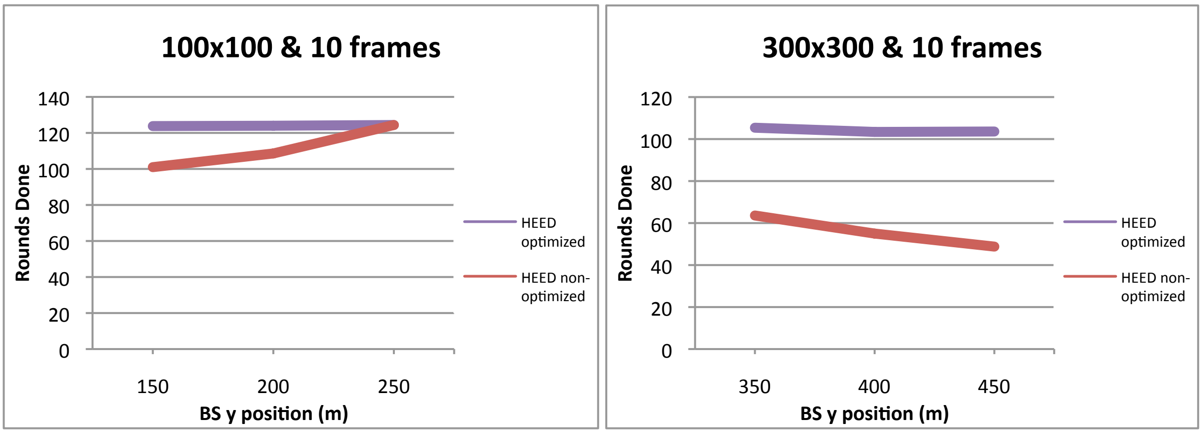


Figure 6.15. HEED optimized and HEED non-optimized results with 10 frames.

Finally, both approaches have been simulated with a steady phase composed by 20 frames as is shown in figure 6.16. In this case the behaviours are still the same as previously. The optimized HEED shows a constant behaviour in all sorts of conditions. Also, it is noticeable that the farther the BS is to the closest node the more the optimized HEED outperforms the non-optimized approach.

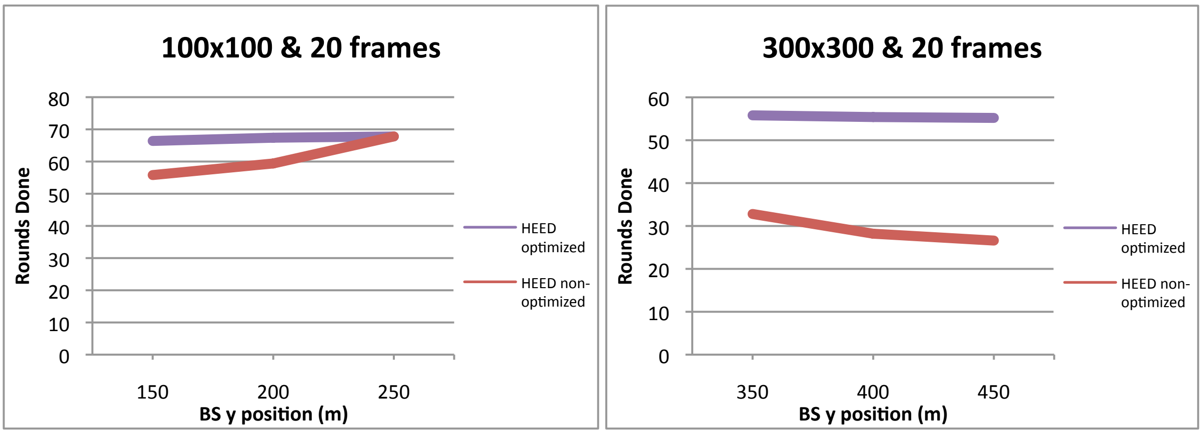


Figure 6.16. HEED optimized and HEED non-optimized results with 20 frames.

If figures 6.14, 6.15 and 6.16 are examined, it can be drawn as a straight conclusion that HEED really shows a constant behaviour and performs well in all kind of conditions. The only effect of making the steady phase longer is a natural reduction in the number of rounds achieved according to the duration of each round. This situation is completely different to the LEACH, LEACH-C and their solar-aware extensions due to HEED selects optimal cluster heads and creates balanced clusters based only on local information, i.e. energy estimation and intra-cluster communication cost.

In figure 6.17 is shown a screenshot of one simulation of the HEED protocol in a certain time. The network depicted in this figure is composed by 100 nodes and has an area of 100x100 meters, specifically, whereas the base station is placed at (50,200) meters. As can be seen in this figure HEED gets balanced clusters through an optimal cluster head selection. On most cases this protocol achieves to set cluster heads only in areas where there are no other cluster heads covering all the regular nodes, i.e. non-cluster head nodes.

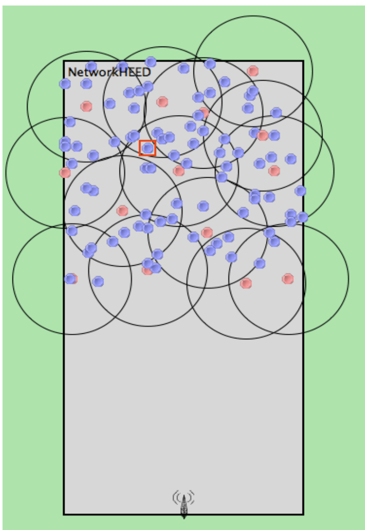


Figure 6.17. Clusters' formation using HEED.

Furthermore, it is remarkable that HEED is able to perform with the same results regardless the distance at which the base station is placed and only decreasing a little bit the overall rounds achieved when the area network is larger and the number of nodes is fixed, i.e. when the node density decreases. This is feasible up to a certain limit set by the features of the node transceiver such as the amount of power levels is able to switch among or the energy consumption the use of a certain power level means.

6.4.2. First node dead

In this subsection is shown a comparison between both approaches of HEED regarding the network lifetime.

Figure 6.18 depicts the case with the shortest steady phase simulated. As in the previous subsection the graph with the smaller area network shows how the results of both protocols get closer the farther the BS is. This is justified another time as the coincidence in the clustering parameters used in both simulations.

In this case the optimized HEED does not show a constant behaviour as previously. Instead of this, the network lifetime achieved is decreasing the farther the BS is to the closest node.

On the other hand, optimized HEED outperforms the non-optimized approach clearly, exceeding between a 250% and more than 300% the rounds achieved by the non-optimized HEED.

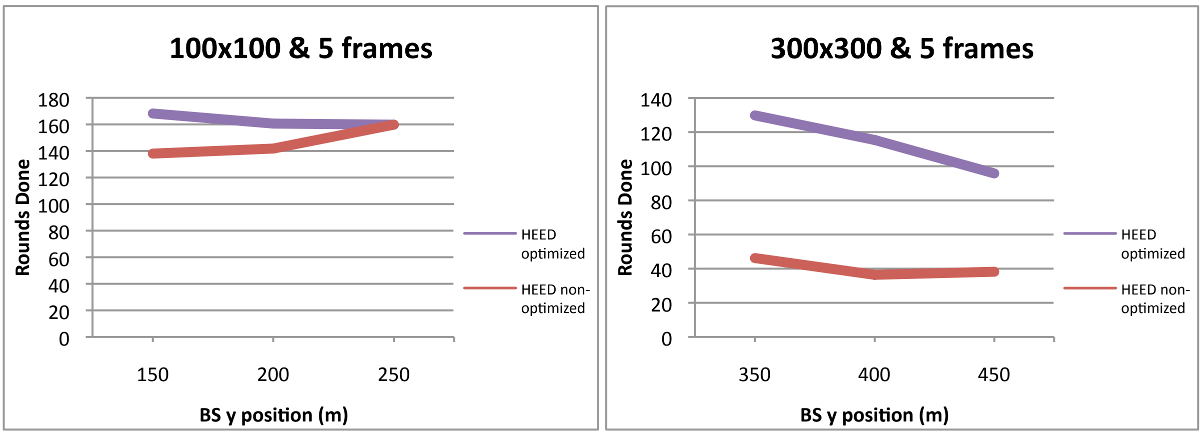


Figure 6.18. HEED optimized and HEED non-optimized results with 5 frames.

In the case of a steady phase composed by 10 frames the comparison between their results is fairly similar to the previous one as is shown in figure 6.19. The only difference is the reduction in the number of rounds achieved by both according to the longer duration of the steady phase. This reduction in the network lifetime is on average the 50% of the previous results.

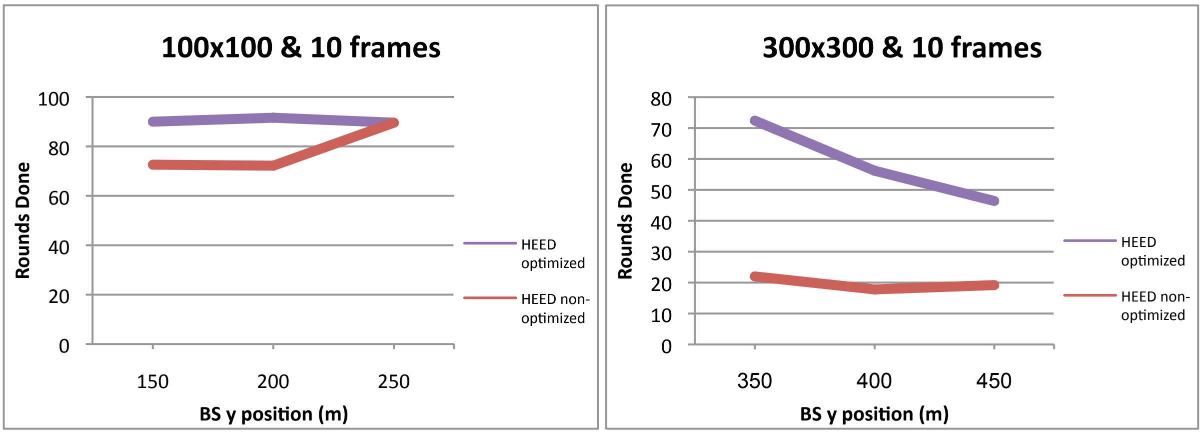


Figure 6.19. HEED optimized and HEED non-optimized results with 10 frames.

Figure 6.20 depicts the network lifetime that both protocols achieved with a steady phase of 20 frames. Another time the graphs show similar results to the last two figures. The optimized HEED gets better results relative to the non-optimized HEED the larger the area network is, however both get closer to each other the farther the BS is to the closest node.

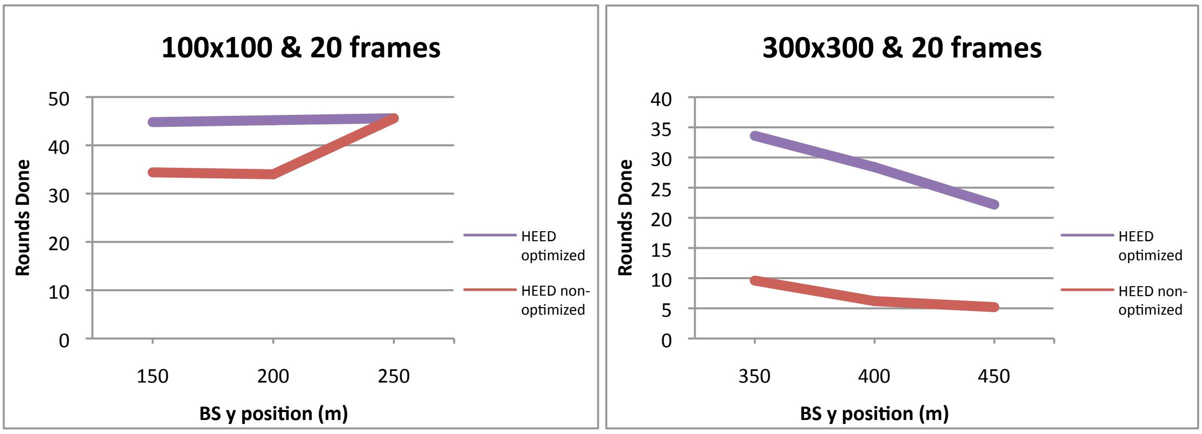


Figure 6.20. HEED optimized and HEED non-optimized results with 20 frames.

Finally, as can be seen analysing figures 6.18, 6.19 and 6.20, both simulations have the same behaviour regardless the duration of the steady phase.

Regarding the optimized HEED, it is noticeable to see how decreases the network lifetime achieved in large networks the farther the base station is. This fact is due to the farther the BS is, the larger distances some nodes have to transmit the data to the BS. Typically these nodes are the closest to the BS and their battery may be depleted faster than the rest of nodes since they are the last hop to relay the data on to the base station.

In the case of the non-optimized HEED this situation is lighter since the inter-cluster communication range is enough large to include the most of the cluster heads of the network unlike the optimized HEED. This causes that all nodes mostly deplete their battery faster and at the same time that it does not depend so much on the distance at which the BS is placed.

6.5. Comparison among different protocols

In this subsection is shown a comparison with the outcomes of the simulations of One-hop, LEACH, LEACH-C and HEED.

Regarding LEACH and LEACH-C solar-aware extensions, it has been considered better not to include them into the comparison since both of them require a solar-collection module as a part of the node, which means a higher cost of the device and hence an approach not feasible in all kind of networks.

In the case of HEED has been utilized the optimized approach, which adjust the clustering parameters to achieve optimal results.

The evaluated results as in the previous subsection show the number of rounds achieved until half of the nodes are already dead or when the first node dies.

6.5.1. Half-dead network

Following is presented a comparison among all the protocols of the rounds achieved until half of the nodes are already dead.

Figure 6.21 shows the results for a steady phase formed by 5 frames in a small and large area network. As can be seen all protocols clearly outperforms the results of One-hop. This direct communication approach only gets results good enough with a base station close to the nodes, since the distance between a certain node and the base station is not excessive assuming that every node is able to communicate directly to the base station, which is a strong assumption.

LEACH and LEACH-C show similar results in all cases. Nevertheless, LEACH-C achieves a higher number of rounds in small area networks unlike in large area networks, where is LEACH which outperforms LEACH-C in results.

Regarding HEED, it is the protocol that gets better results and, the larger the area network and the farther the BS is; the more it outperforms the results of the rest of protocols. Moreover, it shows a really steady behaviour regardless the distance of the BS and the area network.

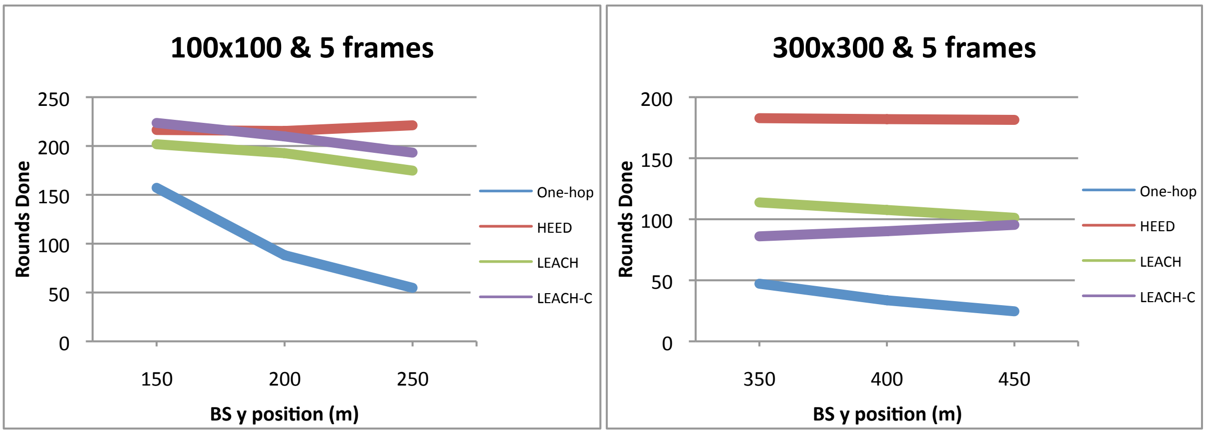


Figure 6.21. Results of all the simulations with 5 frames.

The results of the protocols simulated with a steady phase of 10 frames are shown in figure 6.22. The outcomes of all the protocols are farther similar as the previous ones.

LEACH and LEACH-C show to be even closer to each other than with a steady phase of 5 frames and both achieve approximately the same number of rounds.

As can be observed in this case, HEED gets even better results than with a steady phase of 5 frames relative to the rest of protocols and keeps on showing a constant behaviour in all sorts of conditions.

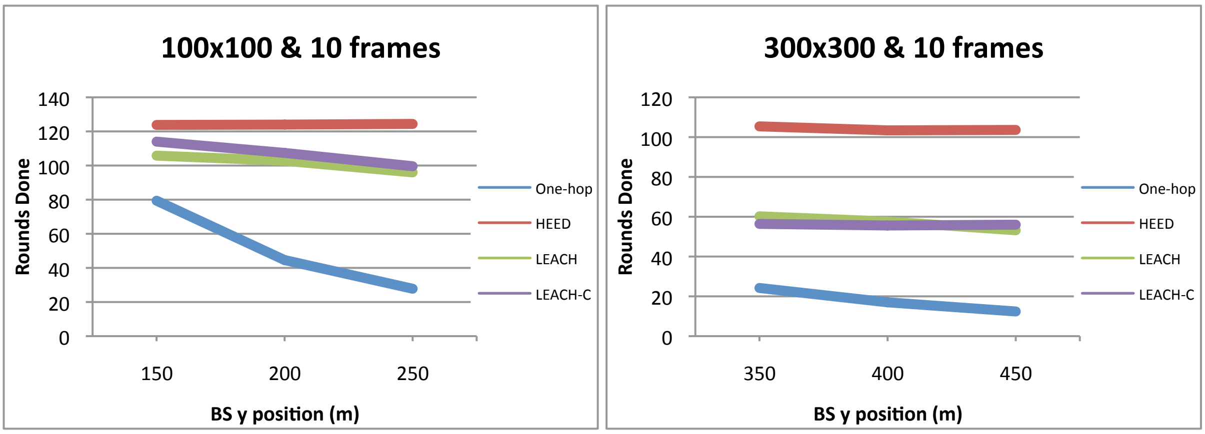


Figure 6.22. Results of all the simulations with 10 frames.

Figure 6.23 depicts the results obtained by all the protocols with a steady phase of 20 frames. As in the last two cases, i.e. with a steady phase of 5 and 10 frames, the results are fairly similar.

With a steady phase of 20 frames LEACH and LEACH-C achieve the same number of rounds and HEED still seems to be constant regardless the distance of the BS to the closest node. All protocols outperform the One-hop results even more than with a steady phase of 5 and 10 frames.

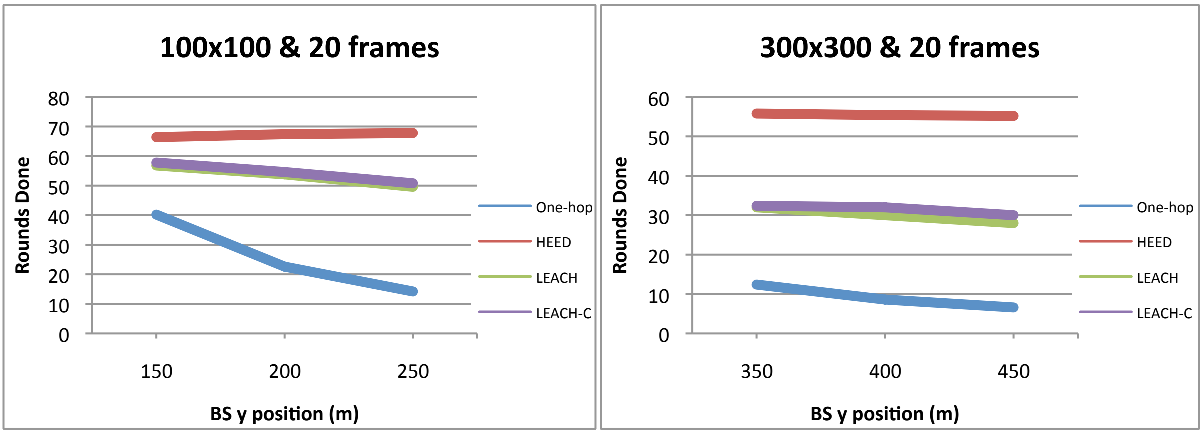


Figure 6.23. Results of all the simulations with 20 frames.

As can be noticed analysing the figures 6.21, 6.22 and 6.23, LEACH, LEACH-C and HEED can be considered as energy-efficient protocols since they easily outperform the outcomes of the One-hop protocol. Furthermore, it is interesting to remark that HEED shows a constant behaviour in all sorts of situations.

Finally, an overview of the results is shown in the table 6.2. It is presented the improvement in percentage of all the compared protocols over the results achieved by the One-hop simulations. For instance, LEACH gets an improvement of 28%, i.e. 202 rounds done, over the 157 rounds achieved by One-hop for an area network of 100x100 meters, a steady phase of 5 frames and the base station placed at (50,150) meters.

Table 6.2. Improvement of rounds achieved in (%) over One-hop results in Half-dead network.

	LEACH				**LEACH-C**				**HEED**			
Area Network	100x100		300x300		100x100		300x300		100x100		300x300	
BS at (m)	**50**	**150**	**50**	**150**	**50**	**150**	**50**	**150**	**50**	**150**	**50**	**150**
5 frames (%)	28	218	142	304	42	250	82	280	37	301	289	624
20 frames (%)	42	257	166	300	45	264	166	328	65	385	366	685

6.5.2. First node dead

Following is presented a comparison of the rounds achieved by all the simulated protocols when the first node dies.

Figure 6.24 depicts the network lifetime obtained by all the protocols with a steady phase composed by 5 frames.

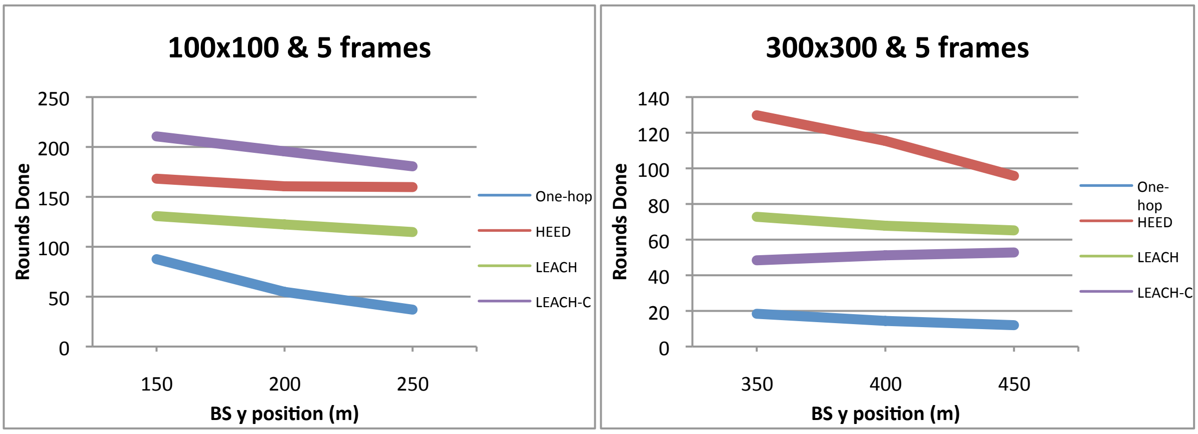
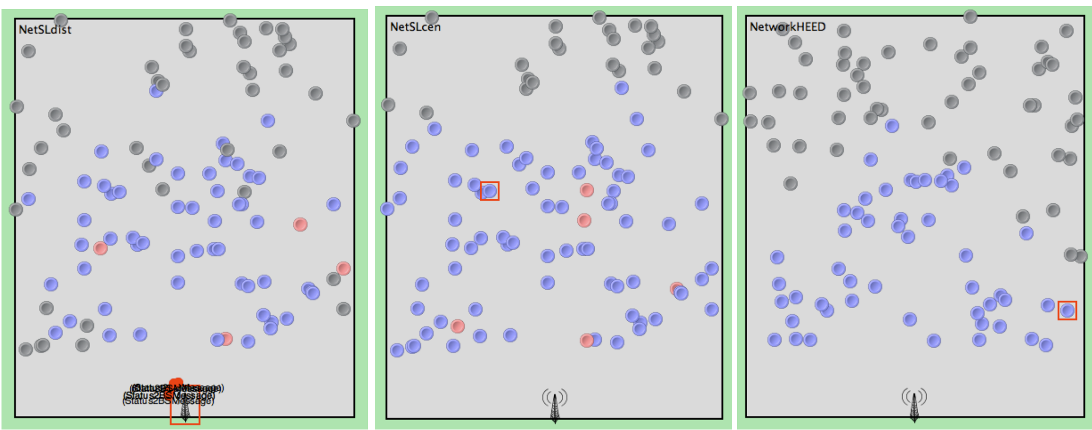


Figure 6.24. Results of all the simulations with 5 frames.

As can be seen in small area networks LEACH-C gets the longest network lifetime. Nevertheless, in a large area network is HEED the protocol that clearly obtains the best result and in this case, even LEACH outperforms the results of LEACH-C. This can be explained as the difference in the operation of each protocol to distribute and select the cluster heads for each round.

In figure 6.25 is shown how are distributed the dead nodes throughout the network for each protocol simulated for an area network of 300x300 meters, the base station placed at (150,350) meters and with a steady phase of 5 frames.



a) b) c)

Figure 6.25. Dead nodes' distribution. a) LEACH-distributed; b) LEACH-centralized; c) HEED

As can be seen in figure 6.25.b), the LEACH-C has dead nodes just in the farthest side of the network to the BS. Despite using the estimated energy of every node to distribute the cluster heads, the fact of being far away from the base station and that some nodes might have been elected several times as cluster heads could cause a complete depletion of the batteries of these nodes. Unlike LEACH-C, under these conditions LEACH (see figure 6.25.a)) and HEED (see figure 6.25.c)) distribute their dead nodes better throughout the whole network.

In the case of a steady phase of 10 frames the outcomes of some protocols vary quite much as is shown in figure 6.26.

HEED protocol gets similar network lifetime than LEACH-C in a small area network but at the same time shows a more constant behaviour regardless the distance of the BS to the closest node. In large area networks HEED clearly outperforms the rest of protocols and LEACH and LEACH-C get similar results. In the latter situation HEED loses its constancy as is explained in section 6.4.2.

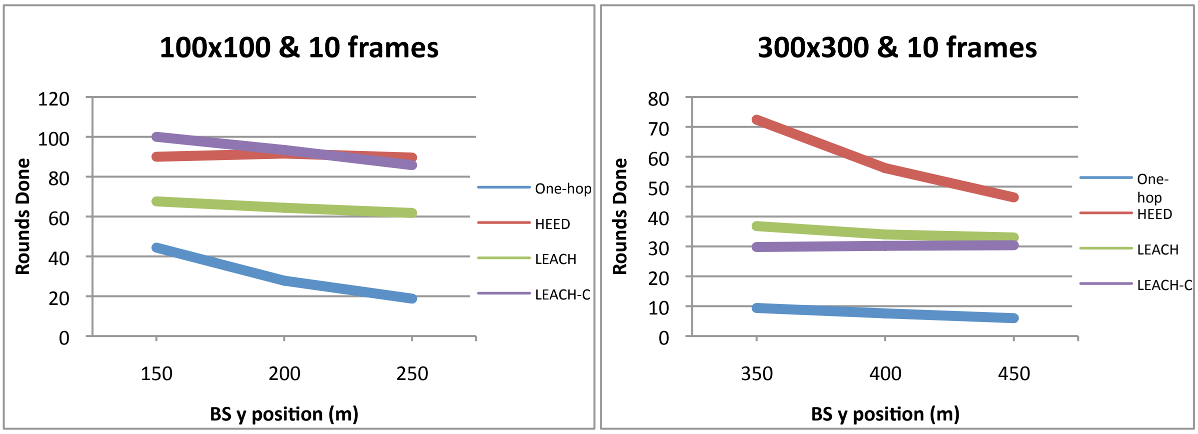


Figure 6.26. Results of all the simulations with 10 frames.

Figure 6.27 depicts the network lifetime achieved by all the protocols with a steady phase of 20 frames.

In this case HEED gets another time the best results in both small and large area networks, though in the former case its results are quite similar to the network lifetime obtained by LEACH-C when the base station is placed close to the network.

The behaviours of the protocols are still the same as explained for steady phases with 5 and 10 frames.

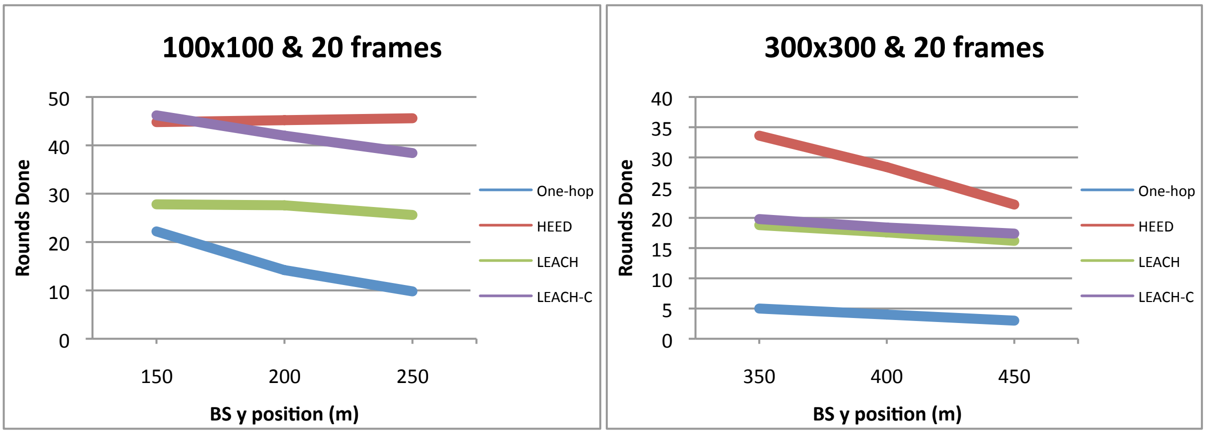


Figure 6.27. Results of all the simulations with 20 frames.

As can be noticed analysing the figures 6.24, 6.26 and 6.27, the longer the steady phase is the more HEED outperforms the results of the rest of protocols. Also regarding to HEED, it is interesting to remark that in small area networks it shows a constant behaviour regardless the distance of the BS to the closest node unlike in large area networks where the network lifetime decreases the farther the BS is.

On the other hand, LEACH-C gets much worse results in large area networks than in small ones compared to the rest of protocols as explained above. However, as can be seen in large area networks LEACH-C improves its results comparing with LEACH the longer the steady phase is.

Finally, an overview of the results is shown in the table 6.3. It is presented the improvement in percentage of all the compared protocols over the results achieved by the One-hop simulations. For instance, LEACH-C gets an improvement of 166%, i.e. 48 rounds done, over the 18 rounds achieved by One-hop for an area network of 300x300 meters, a steady phase of 5 frames and the base station placed at (150,350) meters.

Table 6.3. Improvement of the network lifetime in (%) over One-hop results.

	LEACH				**LEACH-C**				**HEED**			
Area Network	100x100		300x300		100x100		300x300		100x100		300x300	
BS at (m)	**50**	**150**	**50**	**150**	**50**	**150**	**50**	**150**	**50**	**150**	**50**	**150**
5 frames (%)	48	210	305	441	139	389	166	341	90	332	622	700
20 frames (%)	27	160	280	433	109	280	300	466	104	360	580	633

7. Conclusions

In this section is presented an overview of the likely environmental impact this work could have, next some conclusions are drawn and finally is presented a list of future work to complement and improve the goals of this master thesis.

7.1. Environmental impact

Wireless sensor networks are fairly often joined to topics such as environment protection, environmental disasters prevention and environmental study. Well-known instances of these mentioned bonds are forest fires detection, networks to find survivors in disaster zones and military applications in battlefields.

On the other hand, there are also other sorts of applications, which are dedicated to the sensing and measurement of the environmental conditions for researching goals. Therefore, it can be seen the importance of researching in the WSNs field in order to devise and improve applications and systems that could have an straight influence on the environmental evolution in short or long term.

This work deals with initialization and routing protocols for wireless sensor networks. These protocols aim to be energy-efficient in order to elongate the battery lifetime and network lifetime as a result. In most application scenarios the replacement of failed or depleted network nodes is not an option since they are placed in hazardous zones, thus it is extremely important that nodes consume the minimum amount of energy in order to make as long as possible the lifetime of the network, i.e. the time the application is still working properly.

The energy-efficiency of the studied protocols is not the only interesting feature. HEED for instance, also allows deploying less nodes or deploying these nodes in larger area networks, i.e. less node density, and it still gets the same network lifetime and performs in the same manner. Hence this feature could be seen as an extra-saving of expenses in material and reduction of the overall cost of the network deployment. Moreover, using HEED with the appropriate parameters allows a reduction in the amount of physical components used and the environmental impact it involves.

7.2. Summary

After having carried out and finish this work some conclusions can be drawn.

First and foremost, the studied clustering-based routing protocols have proved to be more energy-efficient than the One-hop routing approach, which could be considered the simplest one.

The implementation of HEED that has been done for this work shows to be an energy-efficient protocol and has a balanced clustering formation. The former assertion can be observed in section 6.5, where is clearly the protocol that gets the longest lifetime and the highest number of rounds achieved until half of the nodes are already dead. The later assertion is justified looking at the graphical display of the network while the simulation is running, which always denotes clusters with quite similar number of cluster members and well distributed throughout the network.

Furthermore, HEED also shows to be a good solution to implement in a wide sort of wireless ad-hoc networks due to its constant behaviour that can be seen in the results of the simulations with different parameters presented in section 6.4 and 6.5.

On the other hand, LEACH and LEACH-C are good enough routing protocols for networks with high density of nodes according to the results obtained with the simulations and showed in 6.2 and 6.3. Despite of this assertion, it has to be taken into account that these protocols have strong assumptions such as every node can reach each other including the base station, what it is not feasible in most of cases.

Another interesting conclusion is the improvement achieved using the solar-aware extensions of LEACH and LEACH-C in networks composed by nodes that are provided with a solar-collection module. The results obtained by these protocols show that these protocols are a really desirable option to consider in such a networks since they achieve even the double of network lifetime than the original protocols.

Regarding the OMNeT++ simulator, it is a very powerful framework to implement all sorts of initialization and routing algorithms due to the tools it provides and the ease of programming. At first, the NED language it uses was unknown for me thus I had some problems to start with besides I had to remember about C++ programming, but the website provides to the users a wide range of examples and an useful mailing list and *wiki* for queries and errors found in the users codes.

7.3. Future work

During the time I have been working on this master thesis, some questions and ideas regarding the topic arose to me. Here are presented some of these suggestions for improvements on the work.

There are some suggestions regarding HEED, for instance. This is certainly the most complex protocol implemented and simulated in this work, which has been implemented from scratch, and there are parts of the code that could be improved. Some improvement would be to use a synchronization technique such as RBS suggested in [5] or another important upgrade would be to modify the inter-cluster communication, which is suggested in [18].

Another interesting point would be to simulate the different protocols implemented with higher and variable number of nodes and larger area networks to see how they behave with other conditions and parameters.

One step forward would be to implement HEED over the IEEE 802.15.4 standard, which would be the step before to a real network deployment using HEED as the routing protocol. For this purpose, there are already implemented the physical and media access control layer of this standard in the current distribution of the INET simulation model, though they are still in a testing stage.

Finally, it would be of interest to implement other well-known clustering-based routing protocols such as APTEEN in order to compare them and check which of them is more suitable for a certain wireless ad-hoc network.

References

[1] A. Manjeshwar and D. Agrawal, "TEEN: A Routing Protocol for Enhanced Efficiency in Wireless Sensor Networks," In *Proceedings of the 1st International Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing*, San Francisco, CA, USA, April 2001.

[2] W.R. Heinzelman, A. Chandrakasan and H. Balakrishnan, "Energy-Efficient Communication Protocol for Wireless Microsensor Networks," *Proceedings of the 33rd Hawaii International Conference on System Science*, Vol. 2, Jan 2000.

[3] Yuan Ping and B. Y. Wang Hao, "A Multipath Energy-Efficient Routing Protocol for Ad hoc Networks," *Communications, Circuits and Systems Proceedings, 2006 International Conference on*, Vol. 3, pp. 1462-1466, Guilin, January 2007.

[4] A. Manjeshwar and D. Agrawal, "APTEEN: A Hybrid Protocol for Efficient Routing and Comprehensive Information Retrieval in Wireless Sensor Networks," In *Proceedings of the 2nd International Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing*, pp. 195-202, Ft. Lauderdale, FL, April 2002.

[5] O. Younis and S. Fahmy, "HEED: A Hybrid, Energy-Efficient, Distributed Clustering Approach for Ad Hoc Sensor Networks," *IEEE Trans. Mobile Computing*, Vol. 3, No. 4, pp. 366-379, Oct.-Dec. 2004.

[6] Muruganathan, S.D. and Fapojuwo, A.O., "A Hybrid Routing Protocol for Wireless Sensor Networks Based on a Two-Level Clustering Hierarchy with Enhanced Energy Efficiency," *Wireless Communications and Networking Conference, 2008*, pp. 2051-2056, Las Vegas, NV, April 2008.

[7] S. Lindsey and C. Raghavendra, "PEGASIS: Power-Efficient GAthering in Sensor Information Systems," *IEEE Aerospace Conf. Proc., 2002*, vol. 3, 9-16, pp. 1125-30, April 2002.

[8] S. Meguerdichian, S. Slijepcevic, V. Karayan and M. Potkonjak, "Localized Algorithms In Wireless Ad-Hoc Networks: Location Discovery And Sensor Exposure," *Proceedings of MobiHOC*, pp. 106-116, Long Beach, CA, USA, 2001.

[9] Ravi Prakash, "A Routing Algorithm for Wireless Ad Hoc Networks with Unidirectional Links," *Wireless Networks*, Vol. 7, Issue 6, November 2001.

[10] Sungoh Kwon Shroff, N.B., Samsung Electron. Co., Suwon, "Analysis of Shortest Path Routing for Large Multi-Hop Wireless Networks," *Networking, IEEE/ACM Transactions on*, Vol. 17, No. 3, pp. 857-869, November 2008.

[11] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An Application-Specific Protocol Architecture for Wireless Microsensor Networks," *IEEE Transactions on Wireless Communications*, Vol. 1, No.4, pp. 660-670, Oct 2002.

[12] Loscri, V. Morabito, G. Marano, S., “A Two-Levels Hierarchy for Low-Energy Adaptive Clustering Hierarchy (TL-LEACH),” *Vehicular Technology Conference, 2005*, Vol. 3, pp. 1809-1813, September 2005.

[13] Ameer Ahmed Abbasi and Mohamed Younis, “A Survey on clustering algorithms for wireless sensor networks,” *Computer Communications*, Vol. 30, Issue 14-15, October 2007.

[14] OMNET++ Simulation Environment, http://www.omnetpp.org.

[15] T. Voigt, A. Dunkels, J. Alonso, H. Ritter and J. Schiller, “Solar-aware Clustering in Wireless Sensor Networks,” *Proceedings of the Ninth International Symposium on Computers and Communications 2004*, Vol. 2, pp. 238-243, 2004.

[16] T. Murata and H. Ishibuchi, “Performance evaluation of genetic algorithms for flowshop scheduling problems,” *Proc. 1st IEEE Conf. Evolutionary Computation*, Vol. 2, pp. 802-817, June 1994.

[17] P. Agarwal and C. Procopiuc, “Exact and approximation algorithms for clustering,” *Proc. 9th Annu. ACM-SIAM Symp. Discrete Algorithms*, pp. 658-667, Baltimore, MD, Jan. 1999.

[18] Chia-Hung Lin, Ming-Jer Tsai, "A Comment on "HEED: A Hybrid, Energy-Efficient, Distributed Clustering Approach for Ad Hoc Sensor Networks'," *IEEE Transactions on Mobile Computing*, Vol. 5, No. 10, pp. 1471-1472, Oct. 2006.

[19] Jamil Ibriq and Imad Mahgoub, “Cluster-Based Routing in Wireless Sensor Networks: Issues and Challenges,” *Proceedings of the International Symposium on Performance Evaluation of Computer and Telecommunication Systems*, pages 759-766, San Jose, California, USA, July 25-29, 2004.

Escola Politècnica Superior de Castelldefels
UNIVERSITAT POLITÈCNICA DE CATALUNYA

CHEMNITZ UNIVERSITY OF TECHNOLOGY

APPENDIX

TITLE: Initialization algorithms for wireless ad-hoc networks

MASTER DEGREE: Master of Science in Telecommunication Engineering & Management

AUTHOR: Carlos Agreda Ninot

DIRECTOR: Prof. Dr.-Ing. Ulrich Heinkel

DATE: April 26 th 2010

Appendix A

Here are presented some of the changes that have been done in the codes of LEACH, LEACH-C and their respective solar-aware extensions found in [14] due to the migration process from OMNeT++ 3.0 to version 4.0.

A.1. LEACH and solar-aware extension

Once the steps listed in the migration file provided by [14] and later the INETMANET distribution are finished, it is necessary to change manually some functions and parameters that can not be done automatically.

These changes include the connections between nodes just as between nodes and the base station, some parameters regarding the simulation time or some important changes in the NED language, for instance.

In the figure A.1 it is presented the *initNodes()* function of the original code found in [14]. The code in bold letter is the part that has been deleted or changed since they are functions or parameters that have been deprecated from version 3.0 to 4.0.

```
// making the gates on the fly
void Node::initNodes()
{
    int i;
    cModule *parent = parentModule();
    cModule *mod;
    cModule *myMod;          // this is my Module
    int numNodes;
    int ritems;

    numNodes = (int) parent->par("numNodes");
    ev << "numNodes is: " << numNodes << " energy: " << energy << "\n";

    for (i = 1; i <= simulation.lastModuleIndex(); i++)
    {
        int x, y, id;
        //scan the simulation module vector
        mod = (cModule *) simulation.module(i);
        // check for nodes in transmission range
        if ((strcmp(mod->name(), "node") == 0) || (strcmp(mod->name(), "bs") == 0))
        {
            if (strcmp(mod->name(), "node") == 0)
            {
                id = ((Node *) mod)->myId;
                x = ((Node *) mod)->xpos;
                y = ((Node *) mod)->ypos;
                nodePtr[id] = ((Node *) mod);
            }
            if (strcmp(mod->name(), "bs") == 0)
            {
                id = ((BS *) mod)->myId;
                this->bsId = id;
                x = ((BS *) mod)->xpos;
                y = ((BS *) mod)->ypos;
                this->bsDist = (x / 10 - this->xpos / 10) * (x / 10 - this->xpos / 10)
                               + (y / 10 -  this->ypos / 10) * (y / 10 - this->ypos / 10);
            }
            if (id != this->myId && id == this->bsId)
            {
                // CONNECTIONS
                cGate *g;
                char gName[32];
                int items;
```

```
            if ((((this->ypos - ypos) * (this->ypos - ypos))
                        + ((this->xpos - xpos) * (this->xpos - xpos))) < 2500 || id == this->bsId)
            {
               items = this->gatev.items();
               ritems = mod->gatev.items();

               // make new gate here
               sprintf(gName, "O_%d", id);
               g = new cGate(gName, 'O');
               this->gatev.addAt(items, g); // position, element
               g->setOwnerModule((cModule *) this, items);

               // make new gate at other side
               sprintf(gName, "I_%d", this->myId);
               g = new cGate(gName, 'I');
               mod->gatev.addAt(ritems, g); // position, element
               g->setOwnerModule((cModule *) mod, ritems);

               //CHANNEL

               cLinkType *etere = findLink("etere");
               connect((cModule *) this, items, (cLinkType *) etere, (cModule *) mod, ritems);
               //draw the link
               g = this->gate(items);
            }
         }
      }
   }
   // send energy and solar status to BS
   ev << "call send2BS\n";
   send2BS(this->myId, this->bsId, this->energy, this->myStatus, this->myCluster, this->xpos, this->ypos);
}
```

Figure A.1. Original code of *initNodes()* function.

All the code regarding the connections between nodes and the base station with themselves has been moved into the NED file that defines the network. Obviously, all the connections can still be programmed in C++, i.e. in the simple modules. Even though in the case of LEACH, where it is assumed that every node can reach all others including the base station, this implementation can be done in the NED file, which deals with the initialization of all the simple modules and connections and that is called the network file. This manner of programming makes sense since all the connections in LEACH will not change in the course of the network lifetime.

An example of one of these NED files is the LEACH network file shown in figure A.2. In this picture it is emphasised in bold letter the part of the code that defines the connections, which is called *connections allowunconnected*. This part of the code included in the NED file replaces the old code commented on figure A.1 that sets the connections between all the nodes and between the nodes and the base station.

```
package src;

import src.NodeSLdist;
import src.BsSLdist;

//
// Solar LEACH network
//
network NetSLdist
{
   parameters:
      int numNodes = default(100);
      int trRange = default(190);                    // @unit(m)
```

```
        int rounds = default(500);
        int frames = default(5);                              // 5 TDMA frames per round
        int solarOn = 0;
        int sunDuration = default(500);
        int sunNodes = int(numNodes/10);
        double sRand = default(11111);
        int xMax = default(300);                         //@unit(m)
        int yMax = default(300);                         //@unit(m)
    submodules:
        bs: BsSLdist {
            parameters:
                id = 1;
                randInit = sRand;
                xpos = xMax/2;
                numNodes = numNodes;
                ypos = 350;
            gates:
                in[numNodes];
                out[numNodes];
        }
        node[numNodes]: NodeSLdist {
            parameters:
                energy = intuniform(499900,500000);               // 0.5 Joules
                sunstart = uniform(20,(numNodes/sunNodes)*sunDuration);
                id = index+2;
                xpos = intuniform(0, xMax);
                ypos = intuniform(0, yMax);
                @display("is=vs");
            gates:
                in[numNodes+1];
                out[numNodes+1];
        }
    connections allowunconnected:
        for i=0..numNodes-1, for j=0..numNodes-1 {
            bs.out[i] --> { @display("ls=white,0"); } --> node[i].in[0] if (j==0);
            bs.in[i] <-- { @display("ls=white,0"); } <-- node[i].out[0] if (j==0);
            node[i].out[j+1] --> { @display("ls=white,0,d"); } --> node[j].in[i+1];
        }
}
```

Figure A.2. LEACH network file (NED language).

The NED language has suffered a lot of modifications since the version 3.0. Some key works have been included and also the manner to sort out the different parts of the code has changed. Figure A.3 depicts the network file of the original LEACH for the OMNeT++ 3.0 version.

```
import "node";
import "bs";

//
// Solar LEACH network
//
module Solar
    parameters:
        numNodes,
        trRange,
        rounds,
        frames,
        solarOn,
        sunDuration,
        sunNodes,
        sRand,
        xMax,
        yMax;
```

```
    submodules:
        bs: BS;
            parameters:
                id = 1,
                randInit = sRand,
                xpos = xMax/2,
                numNodes = numNodes,
                ypos = 1750;
        node: Node[numNodes];
            parameters:
                energy = intuniform(499900,500000),
                sunstart = intuniform(20,(numNodes/sunNodes)*sunDuration),
                //energy = energy,
                id = index+2,
                xpos = intuniform(0, xMax),
                ypos = intuniform(0, yMax);
    connections nocheck:
endmodule

//
// Instantiates the network
//
network solar: Solar
endnetwork
```

Figure A.3. LEACH network file for version 3.0.

If figure A.2 and figure A.3 are compared between each other it can be seen the changes that were necessary to fulfil the requirements of version 4.0.

A.2. LEACH-C and solar-aware extension

The steps followed have been exactly the same as for LEACH and Solar-aware LEACH.

Firstly, it has been necessary to follow both migration files and later to change manually all the deprecated parameters and functions to adapt the code to version 4.0.

Furthermore the original code was incomplete and there were some function that had to be finished or implemented from zero. In figure A.4 is shown the *handover* function that was completely done from scratch since in the original code there was only the name of the function.

```
// HANDOVER function
void NodeSLcen::sendNewHead(int newHead, int lastHead)
{
    ClusterHeadMessage *cmsg = new ClusterHeadMessage();
    ev << "send messages to cluster nodes announcing new cluster head\n";
    cmsg->setProto(CL_TOHEAD);          // XXX new name (to all now)
    cmsg->setSrcAddress(lastHead);
    int dist = 0;

        for (int rec = 2; rec < this->nrNodes + 2; rec++)
        {
                dist = 0;
                NodeSLcen *nPtr = (NodeSLcen *) nodePtr[rec];
                if (nPtr->headId == lastHead)
                {
                    cmsg->setCHead(rec, newHead);
                    cmsg->setDestAddress(rec);
```

```
                if (findGate("out",rec-1) > -1)            //# bsId (or rec)-1 = 0
                {
                    if (this->gate("out",rec-1)->isConnected())
                    {
                            ev << this->myId << ": announce new CH for: " << rec << "\n";
                            send((ClusterHeadMessage *) cmsg->dup(),"out",rec-1);
                            dist = (nPtr->xpos - this->xpos) * (nPtr->xpos - this->xpos) +
                                            (nPtr->ypos - this->ypos) * (nPtr->ypos - this->ypos);
                            this->energyTransmit(25 * 8, dist);
                        }
                                            }
                                    }
            }
    if (!cmsg->isScheduled())
            delete(cmsg);
}
```

Figure A.4. Handover function for Solar-aware LEACH-C.

Appendix B

In this chapter are presented the results obtained by each protocol in all the simulations done. For each protocol there are two different tables, the first one regarding the rounds achieved until half of all the nodes are dead and the second one when the first node dies, i.e. the network lifetime.

In each blue row, i.e. where the number of frames appears, it is shown the average of the 5 different simulations done with the same parameters and conditions.

B.1. One-hop tables

In this section are shown the results for the One-hop protocol.

Half-dead network

	ROUNDS DONE								
	One-hop								
Area network	100x100			200x200			300x300		
BS ypos (m)	150	200	250	250	300	350	350	400	450
5 frames	157,2	88,2	54,8	77,4	50	34,6	47,2	33,6	24,6
	148	84	53	90	57	38	48	35	25
	157	88	55	79	52	36	46	32	24
	178	97	59	73	48	33	42	30	22
	152	86	53	69	45	32	52	36	26
	151	86	54	76	48	34	48	35	26
10 frames	79,4	44,6	27,8	39,2	25,4	17,6	24,2	17	12,4
	75	42	27	46	29	20	25	18	13
	79	45	28	40	26	18	24	16	12
	90	49	30	37	24	17	21	15	11
	77	43	27	35	23	16	27	18	13
	76	44	27	38	25	17	24	18	13
20 frames	40,2	22,6	14,2	19,8	13,2	9	12,4	8,6	6,6
	38	21	14	23	15	10	13	9	7
	40	23	14	20	14	9	12	8	6
	45	25	15	19	12	9	11	8	6

	39	22	14	18	12	8	14	9	7
	39	22	14	19	13	9	12	9	7

First node dead

	FIRST NODE DEAD								
	One-hop								
Area network	100x100			200x200			300x300		
BS ypos (m)	150	200	250	250	300	350	350	400	450
5 frames	87,6	54,8	37	35,2	25,6	19,6	18,4	14,4	12
	87	55	37	36	26	20	19	15	12
	84	53	36	33	24	19	19	15	12
	90	56	38	37	27	20	18	14	12
	90	55	37	34	25	19	18	14	12
	87	55	37	36	26	20	18	14	12
10 frames	44,4	27,8	18,8	18,2	13,2	10	9,4	7,6	6
	44	28	19	19	13	10	10	8	6
	42	27	18	17	13	10	10	8	6
	46	28	19	19	14	10	9	7	6
	46	28	19	17	13	10	9	7	6
	44	28	19	19	13	10	9	8	6
20 frames	22,2	14,2	9,8	9,6	7	5	5	4	3
	22	14	10	10	7	5	5	4	3
	21	14	9	9	7	5	5	4	3
	23	15	10	10	7	5	5	4	3
	23	14	10	9	7	5	5	4	3
	22	14	10	10	7	5	5	4	3

B.2. HEED tables

In this section are presented the results for HEED with an optimized version where the cluster radius and inter-cluster communication range are adapted according to the distance at which the base station is placed and a non-optimized version where these two parameters are fixed for each area network regardless the position of the base station.

B.2.1. Optimized HEED

Following are presented both tables for the optimized HEED.

Half-dead network

	ROUNDS DONE								
	HEED optimized								
Area network	100x100			200x200			300x300		
BS ypos (m)	150	200	250	250	300	350	350	400	450
5 frames	216,4	215,4	221,2	192	200	200	182,8	182	181,4
	223	213	222	192	209	201	180	180	178
	213	215	221	193	197	203	184	183	180
	218	221	221	192	202	199	187	183	176
	217	215	221	190	198	203	183	179	190
	211	213	221	193	194	194	180	185	183
10 frames	123,8	124	124,4	109,2	116,2	114	105,4	103,4	103,6
	126	126	126	108	115	112	104	104	105
	123	123	125	108	114	113	104	103	103
	122	125	123	110	116	118	109	101	104
	124	123	124	107	118	115	105	104	104
	124	123	124	113	118	112	105	105	102
20 frames	66,4	67,4	67,8	57,4	62	62,2	55,8	55,4	55,2
	68	67	68	58	62	61	56	56	55
	66	68	67	57	62	62	55	55	55
	67	66	69	58	62	62	56	55	55
	64	68	66	58	62	63	55	57	57

	67	68	69	56	62	63	57	54	54

First dead node

	FIRST NODE DEAD								
	HEED optimized								
Area network	100x100			200x200			300x300		
BS ypos (m)	150	200	250	250	300	350	350	400	450
5 frames	168,2	160,6	159,8	142,6	141	96,4	129,8	115,4	95,8
	172	171	174	140	140	77	141	116	110
	174	160	141	142	135	105	134	128	102
	163	141	152	138	142	111	121	113	76
	174	160	181	146	143	95	125	113	91
	158	171	151	147	145	94	128	107	100
10 frames	90	91,6	89,6	73	74,8	56,2	72,4	56,2	46,4
	93	90	90	72	64	54	69	60	48
	95	93	87	77	74	60	80	61	60
	71	87	93	67	80	54	75	54	39
	94	94	87	69	78	61	67	55	43
	97	94	91	80	78	52	71	51	42
20 frames	44,8	45,2	45,6	37,2	36	29,6	33,6	28,4	22,2
	46	45	50	39	30	21	32	26	22
	49	34	47	37	33	29	32	28	22
	37	48	34	39	38	26	34	25	18
	43	49	48	32	35	36	35	30	27
	49	50	49	39	44	36	35	33	22

B.2.1. Non-optimized HEED

Next can be seen the two tables regarding the results achieved by the non-optimized HEED.

Half-dead network

	ROUNDS DONE								
	HEED non-optimized								
Area network	100x100			200x200					
BS ypos (m)	150	200	250	250	300	350	350	400	450
5 frames	181,6	193,2	221,2	148	133,8	135,2	117,6	101,8	92,4
	185	198	222	151	134	137	112	101	86
	178	189	221	144	139	132	119	100	95
	178	184	221	146	134	134	115	101	92
	184	195	221	146	131	136	119	101	92
	183	200	221	153	131	137	123	106	97
10 frames	101	108,6	124,4	80,8	73,2	72,8	63,6	55	48,8
	102	112	126	81	71	76	61	53	48
	100	109	125	82	72	70	65	54	50
	100	106	123	82	72	72	60	55	46
	101	109	124	79	73	73	67	57	48
	102	107	124	80	78	73	65	56	52
20 frames	55,8	59,4	67,8	42,8	38,6	38	32,8	28,2	26,6
	54	61	68	42	38	41	33	29	25
	55	59	67	42	40	36	32	27	27
	60	59	69	45	38	37	33	29	26
	55	58	66	42	39	37	32	27	27
	55	60	69	43	38	39	34	29	28

First dead node

	FIRST NODE DEAD								
	HEED non-optimized								
Area network	100x100			200x200			300x300		
BS ypos (m)	150	200	250	250	300	350	350	400	450
5 frames	138	141,8	159,8	74,6	75,8	77,6	46,2	36,4	38,2
	126	152	174	79	84	81	52	35	46
	130	138	141	43	82	86	49	39	44
	134	136	152	70	66	75	38	35	23
	150	139	181	94	59	73	45	31	37
	150	144	151	87	88	73	47	42	41
10 frames	72,6	72,2	89,6	34,8	30,2	34,6	22	17,8	19,2
	69	79	90	36	41	32	23	20	24
	71	74	87	28	16	41	33	21	24
	77	65	93	27	23	42	18	13	10
	67	74	87	38	38	21	24	20	19
	79	69	91	45	33	37	12	15	19
20 frames	34,4	34	45,6	17,8	15,6	18,6	9,6	6,2	5,2
	32	33	50	22	15	12	10	5	7
	29	32	47	17	16	22	10	5	3
	37	35	34	11	17	18	8	8	4
	38	34	48	17	16	21	12	6	6
	36	36	49	22	14	20	8	7	6

B.3. LEACH

Here are presented the rounds achieved by LEACH and Solar-aware LEACH until half of the nodes are dead and when the first node dies.

B.3.1. LEACH

Next are shown the tables regarding LEACH.

Half-dead network

	ROUNDS DONE								
	Leach Distributed (LEACH)								
Area network	100x100			200x200			300x300		
BS ypos (m)	150	200	250	250	300	350	350	400	450
5 frames	201,8	192,6	174,8	153,1	143,6	129,9	113,8	107,6	101,2
	108	99	91	80	75	68	113	108	98
	104	102	93	82	77	69	114	104	100
	108	103	91	83	78	69	117	112	103
	105	102	92	78	72	68	116	109	104
	106	101	93	80	76	68	109	105	101
10 frames	105,8	102,8	96	81,6	77	70,8	60,2	57,6	53,2
	105	102	97	81	76	69	60	57	51
	105	103	96	83	80	73	61	57	54
	105	102	97	83	80	71	61	58	55
	106	104	94	80	72	69	60	58	54
	108	103	96	81	77	72	59	58	52
20 frames	56,8	53,8	49,6	42,2	40	37	32	30	28
	56	53	49	42	39	36	31	30	27
	55	51	49	43	41	38	32	30	28
	56	53	51	42	40	37	33	31	29
	59	56	49	42	40	37	32	29	28
	58	56	50	42	40	37	32	30	28

First dead node

	FIRST NODE DEAD								
	Leach Distributed (LEACH)								
Area network	100x100			200x200			300x300		
BS ypos (m)	150	200	250	250	300	350	350	400	450
5 frames	130,7	122,4	114,8	96,5	91,9	82,1	72,8	67,8	65,2
	78	70	63	49	45	37	67	67	64
	71	67	62	52	49	49	79	72	66
	67	65	66	49	49	42	71	68	67
	66	54	62	49	48	39	74	67	66
	62	66	49	55	51	49	73	65	63
10 frames	67,6	64,4	61,8	51,6	49,6	45,4	36,8	34	33
	67	73	66	49	48	41	38	33	30
	74	60	62	55	49	49	37	35	35
	67	67	67	49	49	45	35	33	33
	66	65	65	49	49	43	35	33	33
	64	57	49	56	53	49	39	36	34
20 frames	27,8	27,6	25,6	24,6	23,4	21,8	18,8	17,6	16,2
	23	30	28	22	22	21	16	16	16
	34	30	25	24	24	21	21	19	17
	23	23	23	25	23	23	20	17	14
	29	25	24	26	23	21	18	18	17
	30	30	28	26	25	23	19	18	17

B.3.2. Solar-aware LEACH

Following are shown the tables regarding Solar-aware LEACH.

Half-dead network

	ROUNDS DONE								
	Solar Leach Distributed								
Area network	100x100			200x200			300x300		
BS ypos (m)	150	200	250	250	300	350	350	400	450
5 frames	301	284	264,8	211,2	199,2	186,8	138,4	135,4	126,8
	301	288	265	217	199	190	129	131	116
	295	273	260	207	199	189	150	139	130
	304	286	263	219	204	187	147	141	137
	301	279	266	206	194	183	139	136	128
	304	294	270	207	200	185	127	130	123
10 frames	140,6	133,6	128,6	106	98,8	94,2	73	69,8	65,8
	139	136	129	107	102	95	73	69	64
	144	140	134	105	98	97	78	73	71
	139	128	124	114	103	97	72	72	66
	142	126	124	103	98	91	74	72	71
	139	138	132	101	93	91	68	63	57
20 frames	67	64,2	60	49,2	47	44,6	34,8	32,8	31,8
	64	64	59	45	46	45	32	33	30
	70	64	59	53	50	49	33	30	29
	64	63	64	53	51	44	37	34	34
	67	67	61	47	44	43	37	33	32
	70	63	57	48	44	42	35	34	34

First dead node

	ROUNDS DONE								
	Solar Leach Distributed								
Area network	100x100			200x200			300x300		
BS ypos (m)	150	200	250	250	300	350	350	400	450
5 frames	301	284	264,8	211,2	199,2	186,8	138,4	135,4	126,8
	301	288	265	217	199	190	129	131	116
	295	273	260	207	199	189	150	139	130
	304	286	263	219	204	187	147	141	137
	301	279	266	206	194	183	139	136	128
	304	294	270	207	200	185	127	130	123
10 frames	140,6	133,6	128,6	106	98,8	94,2	73	69,8	65,8
	139	136	129	107	102	95	73	69	64
	144	140	134	105	98	97	78	73	71
	139	128	124	114	103	97	72	72	66
	142	126	124	103	98	91	74	72	71
	139	138	132	101	93	91	68	63	57
20 frames	67	64,2	60	49,2	47	44,6	34,8	32,8	31,8
	64	64	59	45	46	45	32	33	30
	70	64	59	53	50	49	33	30	29
	64	63	64	53	51	44	37	34	34
	67	67	61	47	44	43	37	33	32
	70	63	57	48	44	42	35	34	34

B.4. LEACH-C

In this section are shown the results obtained by LEACH-C and Solar-aware LEACH-C like the sections presented above.

B.4.1. LEACH-C

Next are presented the tables regarding LEACH-C.

Half-dead network

	ROUNDS DONE								
	Leach Centralized (LEACH-C)								
Area network	100x100			200x200			300x300		
BS ypos (m)	150	200	250	250	300	350	350	400	450
5 frames	223,6	209,8	193,2	158,6	154,6	147	86	90,2	95,4
	223	209	193	164	157	147	83	90	87
	222	209	191	164	157	149	93	99	105
	224	210	194	163	158	151	97	97	105
	223	210	194	143	145	138	78	82	89
	226	211	194	159	156	150	79	83	91
10 frames	114	107,4	99,6	83,6	80,8	76,8	56,4	55,6	56
	113	107	100	85	82	77	58	55	53
	114	107	99	85	80	76	60	59	58
	114	108	100	85	83	78	51	55	57
	114	107	99	80	78	74	57	55	56
	115	108	100	83	81	79	56	54	56
20 frames	57,8	54,6	50,8	45,4	42,6	39,4	32,4	32	30
	58	54	51	45	43	40	31	34	28
	57	54	51	46	42	39	32	30	30
	58	55	51	45	42	40	33	34	31
	58	55	50	44	42	39	32	30	30
	58	55	51	47	44	39	34	32	31

First dead node

	FIRST NODE DEAD								
	Leach Centralized (LEACH-C)								
Area network	100x100			200x200			300x300		
BS ypos (m)	150	200	250	250	300	350	350	400	450
5 frames	210,6	195,6	180,6	122	123,4	129,2	48,4	51,2	52,8
	210	196	180	134	132	137	46	46	49
	210	196	178	134	141	136	51	54	58
	213	190	180	122	125	133	51	56	53
	211	198	181	93	95	104	43	48	47
	209	198	184	127	124	136	51	52	57
10 frames	100	93,4	85,8	63,2	62,2	61,6	29,8	30,2	30,4
	103	93	84	70	65	62	26	27	27
	104	93	87	70	69	60	32	33	33
	92	98	86	62	64	62	33	32	33
	101	91	85	52	47	61	26	27	28
	100	92	87	62	66	63	32	32	31
20 frames	46,2	42	38,4	32	28	25,2	19,8	18,4	17,4
	44	43	39	31	29	27	19	17	13
	47	43	41	35	29	23	19	18	20
	47	40	39	33	28	26	20	21	19
	46	44	35	30	26	26	19	17	17
	47	40	38	31	28	24	22	19	18

B.4.2. Solar-aware LEACH

Next are presented the tables regarding Solar-aware LEACH-C.

Half-dead network

	ROUNDS DONE								
	Solar Leach Centralized								
Area network	100x100			200x200			300x300		
BS ypos (m)	150	200	250	250	300	350	350	400	450
5 frames	394,4	371	359,8	240	255,4	227,6	135,4	126,2	124,2
	413	344	341	248	269	238	118	124	107
	384	365	374	237	268	247	147	123	119
	380	390	349	222	264	209	132	110	153
	405	376	369	268	240	207	155	123	125
	390	380	366	225	236	237	125	151	117
10 frames	199,8	200,8	187,6	144	134	116,8	82,6	78,4	73
	199	184	184	147	143	117	77	73	69
	213	201	201	160	135	118	92	66	70
	222	164	187	139	132	124	88	79	68
	174	231	201	140	135	99	86	85	62
	191	224	165	134	125	126	70	89	96
20 frames	97,4	86,8	75,8	69,6	65,4	50,6	49,2	44,2	38,2
	101	87	76	71	61	57	38	33	30
	101	97	74	60	62	40	56	60	41
	88	101	80	76	76	62	52	50	39
	92	72	77	75	69	49	49	43	41
	105	77	72	66	59	45	51	35	40

First dead node

	FIRST NODE DEAD								
	Solar Leach Centralized								
Area network	100x100			200x200			300x300		
BS ypos (m)	150	200	250	250	300	350	350	400	450
5 frames	340,2	311,4	271	214,2	209	167,4	104,8	88,2	84,8
	358	237	285	234	231	203	90	96	71
	321	351	318	210	238	109	107	90	94
	343	347	239	185	218	162	96	71	92
	339	350	283	235	185	182	120	76	64
	340	272	230	207	173	181	111	108	103
10 frames	145,4	107,2	90	67,8	71,4	58,4	57,4	53,4	30
	109	148	32	95	121	77	50	55	22
	182	142	59	27	43	91	74	51	20
	187	101	123	114	117	24	59	30	43
	117	114	157	29	49	29	56	66	25
	132	31	79	74	27	71	48	65	40
20 frames	59,6	43,8	31,4	29,6	24,4	18,8	25,4	19,2	16,6
	82	44	30	28	23	18	21	22	20
	82	47	47	19	18	17	13	21	18
	30	58	29	67	46	20	32	28	15
	46	40	23	23	23	17	15	15	15
	58	30	28	11	12	22	46	10	15